P.S.I.

PAR
JACQUES
DECONCHAT

P.S.I.

102 PROGRAMMES POUR

MO6, TO8

ET TO9+

CONNAISSEZ-VOUS LA COLLECTION THOMSON CHEZ P.S.I. ?

Initiation

— La découverte des MO5 et TO7/70 - Maurice Charbit et Dominique Schraen
— Exercices en Basic pour MO5 et TO7/70 - Maurice Charbit et Dominique Schraen
— MO5 et TO7/70 pour tous - Jacques Boisgontier et Sophie Brébion

Programmation en Basic

— MO5 et TO7/70 en famille - Jean-François Sehan
— Super jeux MO6, TO8 et TO9+ - Jean-François Sehan

Éducation

— MO5 et TO7/70 pour tout petits - Daniel Nielsen
— MO5 et TO7/70 à l'école - Daniel Nielsen
— MO5 et TO7/70 pour réussir en CE2 - Daniel Nielsen
— MO5 et TO7/70 pour réussir en CM1 - Daniel Nielsen
— Destination collège (CM2) - Daniel Nielsen
— Maths et Thomson en sixième - Jacques Deconchat
— Français et Thomson en sixième - Jacques Deconchat et Gisèle Sergeant
— Profs assistance sur Apple II et Thomson - Daniel Nielsen

Maîtrise du Basic

— Basic MO5 et TO7/70, méthodes pratiques - Jacques Boisgontier
— Le Basic des MO5 et TO7/70 - Gilles Blanchard
— Basic plus, 80 routines sur MO5 et TO7/70 - Michel Martin

Système et assembleur

— Assembleur et périphériques des MO5 et T07/70 - Pierre Brandeis et Frédéric Blanc
— Clefs pour MO5 - Gilles Blanchard
— Clefs pour TO7/70 - Gilles Blanchard

Pour tout problème rencontré dans les ouvrages P.S.I.
vous pouvez nous contacter au numéro ci-dessous :

Numéro Vert/Appel Gratuit en France

05 21 22 01

(Composer tous les chiffres, même en région parisienne)

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© *Éditions du P.S.I. - 5, place du Colonel-Fabien - 75491 PARIS Cedex 10*
1986

ISBN 2-86595-393-9

JACQUES
DECONCHAT

102 PROGRAMMES POUR MO6, TO8 ET TO9+

P.S.I.
EDITIONS DU P.S.I.
1986

SOMMAIRE

Avant-propos 8
Organisation de l'ouvrage 9

Niveau 1

Instructions élémentaires 12
Le nombre mystérieux 18
Cube 20
Chaud, froid 22
Les allumettes 24
Allumettes à plusieurs 26
Calendrier perpétuel 28
Bataille navale 30
Golf 32
Solfège 34
Le jeu du 21 36
Pair - Impair 38
Devine un chiffre 40
Le jeu de Wytthof 42
Rebonds 44
Jackpot 46
Roulette 48
Couloirs 50
Echiquier 52
Le jeu de la fève 54
Jeu de l'oie aléatoire 56

Niveau 2

Dimensionnement d'un tableau - Recherche d'un élément dans un tableau et dans une chaîne 60
Dé truqué 62
Visions 64
421 66
L'électricien fou 68
La course au 20 70
Musique 72
Poker 74
Le jeu de saute-mouton 76

Lettres 78
Machinamot 80
Espion 82
Accrochez les wagons 84
Tirage d'un dé truqué 86
Le jeu de Marienbad 88
Compose 90
Jour de la semaine 92
Le jeu de la vie 94
Tour de cartes 97
L'indésirable 100

Niveau 3

Manipulation des chaînes et graphiques haute résolution 104
Un dé 106
Plus grand, plus petit 108
Le pendu 110
Belote 112
Un taquin de calcul 114
Pendu inversé 116
Nicomaque 118
Course de chevaux 120
A travers 122
Mastermind 124
Idem 126
Horloge 1 128
Reines sur l'échiquier 130
Répétitions 132
Mots 134
Permutations 136
Bridge 138
Hanoï 140
Losanges 142
Mathieu 144
Ordre 146
Le loup et les agneaux 149

Niveau 4

Les techniques d'animation élémentaire 154
Tirage de dés 156
Jeu de loto 158

La bombe	160
Ardoise magique	162
Affichage publicitaire	164
Evasion	166
Chapeaux	168
Combat aérien	170
Dessin de dés	172
Le jeu du missile	174
Le singe	176
Le jeu du 15	178
Adresse	180
Sous-marin	182
Zombies	184
Diago	186
Décomaque	188
Chasse au sous-marin	190
Clavier	192
Le lézard d'Alice	194
Calculatrice	196

Niveau 5

Techniques particulières	201
Smokey	202
Course de voitures	204
Danger spatial	206
Tic Tac Toe	208
Anagrammes	210
Œufs	212
Taquin de lettres	214
Cibles	216
Gouttes	218
Obstacles	220
Captures dans l'espace	222
Anomalie	224
Plusieurs en un	226
Parcours du cavalier	228
Mur de briques	230
Othello	232
Dames	235
Horloge 2	238
Quatre à quatre	241
Echecs	244

Avant-propos

Vous trouverez dans ce livre plus de 100 programmes (essentiellement des jeux), rédigés en Basic Microsoft, pour l'ordinateur individuel MO6. Les programmes sont compatibles avec les 3 versions de Basic : Basic 1.0, Basic 128 et Basic 512. La plupart de ces programmes seront d'ailleurs utilisables sans grande modification sur d'autres ordinateurs utilisant aussi un Basic Microsoft, et ils sont tous utilisables tels quels sur les appareils de la gamme Thomson, TO8 et TO9+.

Les programmes proposés sont très courts, et l'ordre dans lequel ils sont présentés n'est bien évidemment pas arbitraire : il s'agit d'initier le lecteur, à travers la réalisation et l'utilisation de programmes de jeux de bonne qualité, à la connaissance et à la maîtrise progressive de cet outil extraordinaire qu'est un ordinateur individuel.

La brièveté des programmes proposés permettra de les taper facilement, sans risque d'erreur. Vous pourrez bien sûr les utiliser comme base pour la réalisation de jeux plus performants.

Les programmes sont abondamment commentés pour vous aider à comprendre leur fonctionnement, et les pages de présentation vous permettront de retrouver facilement les caractéristiques propres des instructions Basic qu'ils utilisent.

Les programmes n'utilisent pas toutes les particularités d'un appareil comme le MO6, par ailleurs très bien détaillées dans le manuel de base.

Le niveau 5 donne une idée sur l'utilisation de certaines techniques un peu spécifiques, comme le crayon optique, ou les caractères graphiques utilisateur. Ces techniques pourront être reprises avec profit pour améliorer les programmes précédents.

Vous trouverez, à la fin de chaque texte de présentation, quelques suggestions pour des améliorations possibles, ou des modifications, qui pourraient rendre le jeu proposé plus attrayant, plus performant ou plus complet.

Organisation de l'ouvrage

Les jeux sont classés par ordre de difficulté croissante, suivant 5 niveaux :

- **Le niveau 1** permet l'étude des instructions élémentaires du langage Basic : **LET** (optionnel ici), **PRINT, INPUT, IF...THEN, FOR...NEXT, GOTO, GOSUB, ON...GOTO, ON...GOSUB.**

- **Au niveau 2,** on utilisera les instructions de confection et de manipulation des éléments d'un tableau, et de recherche d'un élément dans une chaîne : **DIM, R(I), MID$, READ, DATA.**

- **Au niveau 3,** on utilise la panoplie complète des instructions de traitement des chaînes de caractères, et les instructions permettant le graphisme haute résolution : **MID$, RIGHT$, LEFT$, PSET.**

- **Le niveau 4** fait appel à une instruction très particulière qui facilite la réalisation de jeux interactifs : **INKEY$.**

- Enfin, **le niveau 5** nous fait découvrir des instructions plus spécifiques : définition des caractères graphiques utilisateur, identification d'objets sur l'écran, utilisation du crayon optique : **GR$, SCREEN** (**X, Y**), **POINT** (**X, Y**), **INPEN, PTRIG.**

Chaque jeu est présenté d'une façon très systématique, par une courte description, suivie de l'étude détaillée de chacune des lignes importantes du programme. Après quoi les indications nécessaires au bon fonctionnement du jeu sont données, ainsi que quelques éléments pouvant conduire à des extensions ou des améliorations du jeu proposé.

Le programme, fonctionnant sur le MO6, est ensuite reproduit, ainsi qu'un exemple d'exécution. Les listings ont été obtenus directement d'après les originaux de l'auteur, sur une imprimante thermique, ce qui devrait garantir contre tout risque de disfonctionnement. Les minuscules (obtenues sur MO6 par frappe simultanée de la touche majuscule et de la barre d'espace) ne sont pas utilisées.

Beaucoup de programmes vous demanderont au départ de frapper un entier (ne pas dépasser 1000) : cet entier est utilisé pour initialiser le générateur aléatoire du MO6, à l'aide d'une simple boucle. On pourra redonner le même entier, après un RUN, pour refaire un jeu dans les mêmes conditions que lors de la partie précédente.

Niveau 1

Nom du programme	*Thèmes et particularités*
Le nombre mystérieux	Jeu de déduction, pour un joueur. Plusieurs niveaux.
Cube	Jeu de dessin : perspective sur un écran.
Chaud, froid	Logique et hasard, pour un joueur. Plusieurs niveaux.
Les allumettes	Jeu de réflexion, contre l'ordinateur.
Allumettes à plusieurs	Jeu de réflexion et de stratégie, pour plusieurs joueurs.
Calendrier perpétuel	Confection d'un calendrier, mois par mois.
Bataille navale	Un seul navire à trouver, mais quelques particularités.
Golf	Un parcours de 18 trous, pour un seul joueur.
Solfège	Apprenez le solfège à l'aide de votre ordinateur.
Le jeu du 21	Jeu de hasard, l'ordinateur tient la banque.
Pair - Impair	Un jeu de casino en trois temps (1 ou plusieurs joueurs).
Devine un chiffre	Un jeu très facile, pour un ou plusieurs joueurs.
Le jeu de Wytthof	Un jeu de réflexion assez simple, pour deux joueurs.
Rebonds	Un jeu d'adresse très visuel, pour un joueur.
Jackpot	Jeu de hasard, pour un joueur.
Roulette	Une simulation élégante du jeu très connu.
Couloirs	Un jeu de logique, à découvrir.
Echiquier	Une base graphique pour vos propres créations.
Le jeu de la fève	Un jeu de réflexion, pour deux joueurs.
Jeu de l'oie aléatoire	Un très vieux jeu, revu et corrigé par l'ordinateur, deux joueurs.

Instructions élémentaires

Nous ne retiendrons ici que les instructions fondamentales, ou instructions de programmation ; beaucoup d'autres commandes, ou fonctions particulières, seront utilisées à l'occasion, par exemple :

– ***RND pour obtenir un nombre aléatoire entre Ø* et 1.**

– ***INT pour prendre la partie entière d'un résultat.***

– ***REM pour placer un commentaire.***

– ***END pour terminer un programme.***
etc.

Les instructions fondamentales de la programmation, utilisées dès le niveau 1, et présentes dans pratiquement tous les programmes sont :

LET (optionnelle) ; **PRINT ; INPUT ; GOTO___/; IF___/THEN___/; FOR ... NEXT ; GOSUB ... RETURN.**

soit seulement sept instructions de base (et deux instructions dérivées) que nous allons maintenant examiner plus en détail.

L'INSTRUCTION LET OU INSTRUCTION D'AFFECTATION

C'est l'instruction par excellence des langages de programmation évoluée, puisque son rôle essentiel est de permettre, pour le programmeur, une gestion transparente de la mémoire. Cela signifie, en pratique, que pour ranger une quantité en mémoire, par exemple 12, il n'est pas nécessaire de savoir dans quelle case de la mémoire on va la déposer : il suffit de lui affecter un nom autorisé, et c'est par ce nom que la quantité sera par la suite retrouvée et manipulée.

– **Si la quantité est numérique,** le nom sera une simple lettre de l'alphabet, ou un ensemble de plusieurs chiffres ou lettres, commençant obligatoirement par une lettre.

– **Si la quantité est alphanumérique,** la même règle s'applique pour le nom, qui devra cependant être suivi du symbole $, et la variable devra être écrite entre guillemets.

Exemple :

LET A1 = 12 ; la variable numérique A1 prend la valeur 12

LET NOM $ = "RENOIR" ; la variable alphanumérique NOM$ prend la valeur RENOIR.

Sur le MO6, comme dans la plupart des Basic, le LET est devenu optionnel, aucune ambiguïté n'étant possible, on écrira donc : A = 12 ; NOM$ = "TOTO" ; etc.

L'INSTRUCTION PRINT OU INSTRUCTION DE SORTIE (ECRAN)

C'est une instruction de dialogue, qui permet à la machine d'afficher sur l'écran un texte, le résultat d'un calcul, ou une variable. C'est le seul moyen que possède l'ordinateur individuel pour faire connaître ses résultats. Ainsi, PRINT A1 provoque l'affichage de 12 (si l'on a fait auparavant A1 = 12), PRINT A$ provoque l'affichage de BONJOUR (si l'on a fait A$ = "BONJOUR"). Utilisé après un effacement d'écran, le PRINT provoque l'affichage sur la 1re ligne, en haut, à gauche.

Un certain nombre de paramètres peuvent être utilisés pour obtenir sur l'écran un affichage convenable :

- **PRINT TAB(N)** provoque sur la ligne d'impression courante un affichage commençant à la énième colonne à partir de la gauche (Ø ⩽ N ⩽ 39).

- Un ; (**point-virgule**) en fin de PRINT permet soit d'enchaîner une autre impression sans répéter l'instruction PRINT, soit d'indiquer que la prochaine impression sera, en l'absence d'indication contraire, effectuée exactement à la suite de l'impression précédente.

- La , (**virgule**) provoque une tabulation horizontale automatique prédéfinie.

Lors de la frappe, **l'instruction PRINT** pourra être abrégée en tapant **? (point d'interrogation).** Lors d'une impression, tout nombre est précédé d'un signe ; mais le signe positif n'est pas affiché, un espace précédera donc automatiquement l'impression de tout résultat positif.

L'INSTRUCTION LOCATE

Le MO6 dispose d'une instruction particulière, d'un emploi très souple, qui ***permet de positionner le curseur sur l'écran, en vue de la prochaine écriture.*** Il s'agit de l'instruction LOCATE X, Y (,T), qui sera généralement utilisée sous la forme LOCATE X, Y. X correspond à une tabulation horizontale (entre Ø et 39) et Y à une tabulation verticale (entre Ø et 24). Le paramètre T, facultatif, prendra la valeur Ø si l'on veut faire disparaître la trace du curseur (ce qui sera utile pour certains jeux).

Ainsi, pour écrire le mot BONJOUR à la 5ᵉ colonne de la 8ᵉ ligne, on écrira :

LOCATE 5,8 : PRINT "BONJOUR"

L'INSTRUCTION INPUT OU INSTRUCTION D'ENTREE (CLAVIER)

Cette instruction, qui provoque un arrêt et une attente de l'appareil, ***permettra d'entrer un nombre, ou un texte, qui sera affecté à la variable suivant l'ordre INPUT.*** Ceci est fait en mode conversationnel, c'est-à-dire au fur et à mesure des besoins ; en effet, lorsque le MO6 rencontre une instruction INPUT, il s'arrête et vous rend la main ; on peut alors lui entrer une information de nature convenable (nombre ou texte), et lui signaler que cette entrée est terminée à l'aide de la touche **RETURN,** ce qui lui rendra la main.

Si l'information entrée n'est pas de nature convenable (vous tapez, par exemple, un texte sur un INPUT X, qui attend un nombre), le message **REDO** est affiché, et l'appareil se remet en attente.

Il est possible, comme pour le PRINT, d'afficher des messages avant l'entrée de la variable, et d'utiliser l'instruction LOCATE.

On pourra, par exemple, écrire :

INPUT "VOTRE NOM, S'IL VOUS PLAIT" ; A$

L'INSTRUCTION GOTO OU INSTRUCTION DE RENVOI INCONDITIONNEL

Une des particularités du langage Basic, qui le rend facilement accessible, est que ***chaque ligne de programme doit être précédée d'un numéro,*** ces lignes étant ensuite exécutées par l'ordinateur individuel en séquences, dans le sens des numéros croissants. Pour diverses raisons, il pourra être nécessaire au cours du déroulement d'un programme, de ne pas suivre cet ordre séquentiel très rigide ; on utilisera alors ***l'instruction GOTO, qui permet de détourner le fonctionnement du programme de son cours normal*** quitte d'ailleurs à le reprendre ensuite par un autre GOTO. Le renvoi doit, par ailleurs, impérativement se faire sur un numéro de ligne existant dans le programme (GOTO 2Ø, par exemple).

Pour donner un peu de souplesse à cette instruction de renvoi, il existe une instruction très puissante, qui permet de renvoyer à n'importe quel numéro de ligne, selon la valeur d'une variable numérique : **c'est l'instruction ON___ GOTO___ ;** on écrira par exemple :

ON X GOTO 1ØØ, 2ØØ, 3ØØ et le renvoi sera fait à la ligne 1ØØ
si X vaut 1, à la ligne 2ØØ si X vaut 2, et 3ØØ si X vaut 3.

L'INSTRUCTION IF___/THEN___/OU INSTRUCTION DE TEST

On peut l'utiliser de deux façons : d'abord, en rupture de séquence conditionnelle, sous la forme :

IF <u>condition = vraie</u> THEN <u>numéro de ligne.</u>

C'est l'usage le plus courant de ce test, qui permet de renvoyer à une autre partie de programme, si par exemple X = Ø, ou si A = B, etc. Les **opérateurs logiques NOT, OR** ou **AND** peuvent d'ailleurs être utilisés dans l'écriture de la condition.

On pourra aussi utiliser le test sous la forme :

IF <u>condition</u> THEN <u>instruction</u> ;

l'instruction présentée à la suite de THEN peut être une affectation A = 12, une impression PRINT A, une entrée INPUT B $, ou même un autre test IF___/THEN___/.

On peut par exemple écrire :

IF A > 3 THEN IF B < 7 THEN PRINT A + B

L'INSTRUCTION FOR...NEXT OU INSTRUCTION DE BOUCLES

On est très souvent conduit, en programmation, à vouloir exécuter un nombre entier de fois une certaine suite d'instructions (par exemple, faire avancer un mobile de la case 6 à la case 2Ø, etc.). Plutôt que d'utiliser un test et un renvoi, il sera préférable de faire appel à une ***boucle,*** la variable de boucle prenant ses valeurs dans les entiers ou les décimaux, et le ***pas*** de cette variable (c'est-à-dire la quantité dont elle augmente ou diminue à chaque passage) pouvant aussi prendre ses valeurs dans les entiers ou les décimaux.

Ainsi, si l'on écrit :

```
FOR I =⎫  Ø TO 2Ø STEP 2
       ⎬  Instructions
NEXT I ⎭
```

Les instructions contenues dans la boucle seront exécutées pour I = Ø, pour I = 2, ... et ceci jusqu'à I = 2Ø, soit 11 fois. La variable I peut être utilisée pour faire un calcul dans la boucle, mais ne devra pas être modifiée dans la boucle. On peut, bien sûr, sortir de la boucle, avant I = 2Ø, par un test portant une quantité quelconque.

Si ***le pas de la boucle (STEP)*** n'est pas précisé, il sera automatiquement pris égal à 1.

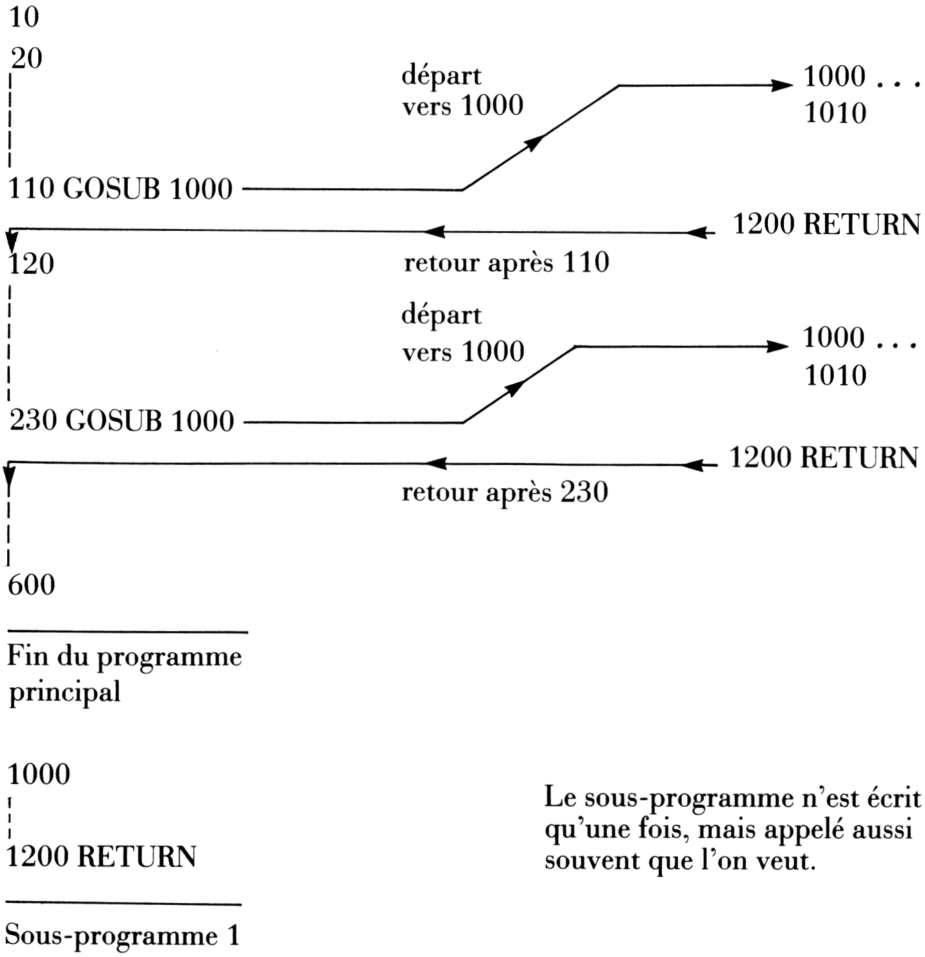
Il est possible d'emboîter plusieurs boucles de noms différents, et d'omettre l'indication de la variable dans le NEXT.

L'INSTRUCTION GOSUB OU INSTRUCTION D'APPEL DE SOUS-PROGRAMME

Lorsqu'une même suite d'instructions doit être exécutée plusieurs fois, et appelée à partir de plusieurs endroits du programme, on évitera de la réécrire plusieurs fois. On la rédige une fois pour toutes, et on la place en dehors du programme proprement dit (en lui donnant, par exemple, un numéro de ligne supérieur au numéro de la dernière ligne du programme). Indépendamment des difficultés que l'on pourra rencontrer au niveau du nom des variables, lorsque l'on sera désireux d'utiliser cette séquence, il suffira de l'appeler par **l'instruction GOSUB numéro de ligne**. Le programme ira dès lors "faire un détour" dans la partie du programme appelé, mais il ne pourra revenir et poursuivre son déroulement normal que si il rencontre dans le programme appelé une instruction **RETURN,** ou instruction de retour.

Voici le schéma théorique de fonctionnement :

Programme principal



Le sous-programme n'est écrit qu'une fois, mais appelé aussi souvent que l'on veut.

Un sous-programme pourra lui-même éventuellement appeler un autre sous-programme, et ainsi de suite.

On peut aussi appeler un sous-programme de façon conditionnelle, en utilisant l'instruction ON___ GOSUB ___.

Par exemple ON X GOSUB 1ØØØ, 2ØØØ, 3ØØØ permettra d'exécuter le sous-programme placé en 1ØØØ si X = 1, en 2ØØØ si X = 2, en 3ØØØ si X = 3.

Le nombre mystérieux

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction pour un joueur.
Difficulté du jeu : facile.

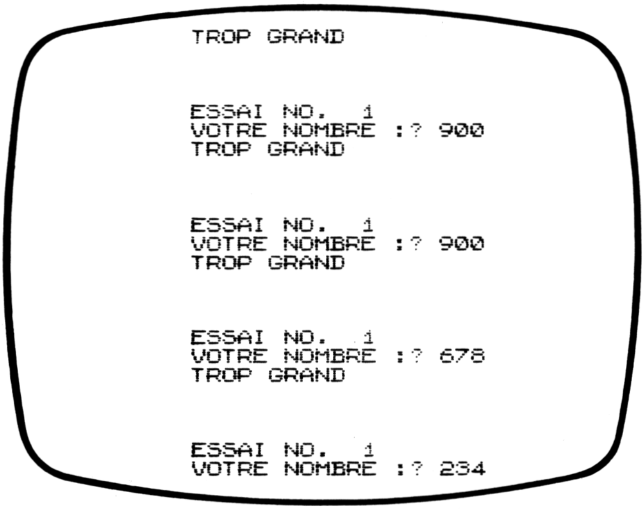
LE JEU :
On demande à l'ordinateur de fabriquer un nombre de n chiffres (avec n compris entre 1 et 7).
On devra ensuite essayer de le découvrir, avec le moins d'essais possible, en recevant les seuls renseignements TROP GRAND ou TROP PETIT.

LE PROGRAMME :
La ligne 5Ø permet de fabriquer le nombre à deviner.
Les entrées sont faites par les lignes 7Ø à 8Ø.
La ligne 9Ø teste si le résultat est exact.
La ligne 1ØØ signale un essai trop petit, et la ligne 11Ø un essai trop grand.
Le compteur d'essais est en ligne 12Ø.
Les lignes 2ØØ à 23Ø s'occupent de l'affichage du résultat, et de la gestion de la partie suivante.

POUR JOUER :
On lancera le programme par RUN.
On obtient un premier affichage NOMBRE DE CHIFFRES auquel on répond par un entier entre 1 et 7, suivi de ENTREE.
On obtient alors l'affichage ESSAI N° 1, suivi de la question VOTRE NOMBRE.
On entre un nombre, et l'on presse ENTREE.
On obtient alors une réponse TROP PETIT, ou TROP GRAND, ou encore, si l'on a trouvé juste un affichage de VOUS AVEZ TROUVE... EN... ESSAIS. Si le résultat n'est pas exact, on revient à ESSAI N° 2, et VOTRE NOMBRE.
En fin de partie, on obtient la question AUTRE PARTIE (O/N) ; on répond O pour faire une autre partie (suivi de ENTREE).
On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra imaginer de faire deviner au joueur autre chose qu'un chiffre (par exemple, un mot, en utilisant l'ordre alphabétique, ou encore autre chose...).

```
10 REM LE NOMBRE MYSTERIEUX
20 CLS
30 INPUT"NOMBRE DE CHIFFRES :";N
40 IF N>7 THEN RUN
50 X=INT(RND*10^N+1)
60 J=1
70 PRINT:PRINT"ESSAI NO. ";J
80 INPUT"VOTRE NOMBRE :";M
90 IF M=X THEN 200
100 IF M<X THEN PRINT"TROP PETIT"
110 IF M>X THEN PRINT"TROP GRAND"
120 J=J+1
130 PRINT:PRINT
140 GOTO 70
200 CLS
210 PRINT"VOUS AVEZ TROUVE ";X;" EN ";J;
" ESSAIS"
220 INPUT"AUTRE PARTIE (O/N)";R$
230 IF R$="O" THEN RUN
```



Cube

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

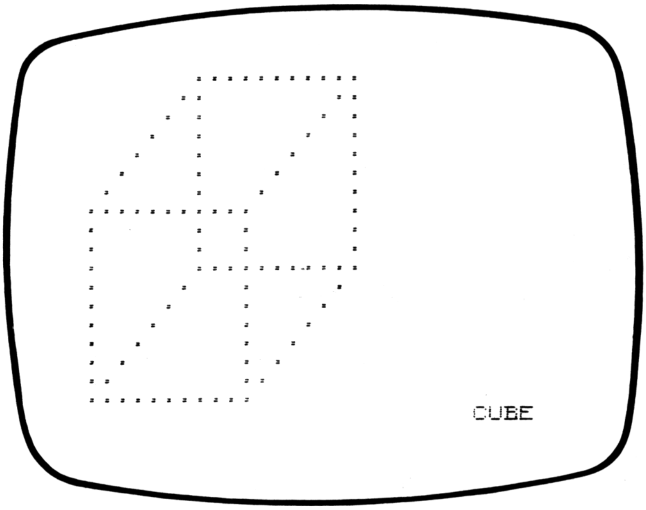
LE JEU :
Le programme proposé se contente de réaliser à l'aide d'instructions PRINT une vue d'un cube en perspective cavalière.
Il pourrait être par exemple incorporé à un jeu basé sur les capacités de reconnaissance des formes géométriques élémentaires.

LE PROGRAMME :
Le dessin du cube est réalisé par l'impression de points, en utilisant deux boucles : la boucle I (3Ø-12Ø), et la boucle I (13Ø-18Ø).
Le nom de l'objet représenté est affiché en ligne 19Ø (ici : CUBE).

POUR JOUER :
Il suffit de lancer l'exécution par RUN, et de regarder.

EXTENSIONS POSSIBLES :
La haute résolution permettra d'avoir un tracé plus précis.
On pourra imaginer un jeu utilisant des parallélépipèdes rectangles de différentes tailles, avec affichage par exemple des 3 côtés, et il faudrait trouver leur volume, etc.

```
10 REM CUBE
20 CLS:COLOR 5,3
30 FOR I=0 TO 9
40 LOCATE 7+I,0:PRINT"."
50 LOCATE I,7:PRINT"."
60 LOCATE 7+I,10:PRINT"."
70 LOCATE I,17:PRINT"."
80 LOCATE 0,7+I:PRINT"."
90 LOCATE 10,7+I:PRINT"."
100 LOCATE 7,I:PRINT"."
110 LOCATE 17,I:PRINT"."
120 NEXT I
130 FOR I=0 TO 7
140 LOCATE I,7-I:PRINT"."
150 LOCATE 10+I,7-I:PRINT"."
160 LOCATE I,17-I:PRINT"."
170 LOCATE 10+I,17-I:PRINT"."
180 NEXT I
190 LOCATE 25,18:COLOR 4,6:PRINT"CUBE"
```



Chaud, froid

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Un des plus vieux jeux du monde, toujours aussi pratiqué, mais cette fois, c'est l'ordinateur qui vous sert de partenaire : il va en effet cacher un objet à un certain emplacement d'une grille dont vous aurez vous-même choisi les dimensions. Pour le trouver, il ne vous donnera que quelques indications sous la forme TU BRULES, ou C'EST FROID, etc.
Vous devez réussir à découvrir l'objet caché, avec le moins d'essais possible (votre score sera affiché en fin de partie).

LE PROGRAMME :
Les lignes 4Ø et 5Ø définissent la position de l'objet.
La ligne 9Ø calcule la distance par rapport aux deux coordonnées que vous avez entrées aux lignes 6Ø-7Ø.
La ligne 1ØØ regarde si vous avez gagné.
Les lignes 12Ø-13Ø normalisent la distance (qui doit avoir un sens, quelles que soient les dimensions de la grille initiale).
La ligne 14Ø effectue le départ, s'il y a lieu, vers l'un des sous-programmes d'affichage.

POUR JOUER :
On lance le programme par l'ordre RUN, puis on introduit les dimensions de la grille (carrée) et l'on frappe ENTREE.
Dès l'affichage suivant (DONNEZ X), on entre la coordonnée X supposée, suivie de ENTREE.
On obtient ensuite l'affichage DONNEZ Y, et l'on entre Y suivi de ENTREE.
On obtient en réponse un texte, qui est variable selon que l'on sera plus ou moins loin de l'objet cherché (cela va de COMPLETEMENT GELE à BRULANT). Lorsque l'on a trouvé, on obtient l'affichage du nombre d'essais effectués.
Pour sortir du programme avant la fin de la partie, il suffit de taper CNT et C.

EXTENSIONS POSSIBLES :
Si l'on dispose de la couleur, il sera intéressant de trouver un moyen plus visuel pour symboliser la distance, par exemple des pavés allant du rouge au bleu, aux emplacements essayés.
On peut aussi travailler dans un espace à trois dimensions, cacher plusieurs objets, etc.

```
10 REM CHAUD,FROID
20 INPUT"DONNEZ UN ENTIER";F
25 FOR I=1 TO F:Y=RND:NEXT I
30 INPUT"DIMENSION DE LA GRILLE :";D
40 X=INT(RND*D+1)
50 Y=INT(RND*D+1)
55 CLS
60 INPUT"DONNEZ X :";A
70 INPUT"DONNEZ Y :";B
80 S=S+1
90 T=ABS(X-A)+ABS(Y-B)
100 IF T=0 THEN 6000
110 PRINT"VOUS ETES ";
120 N=INT(SQR(D)/T)+1
130 IF N>5 THEN N=5
140 IF T<=SQR(D) THEN ON N GOTO 1000,200
0,3000,4000,5000
150 PRINT"COMPLETEMENT GELE"
160 GOTO 60
1000 PRINT"A PEINE TIEDE"
1010 GOTO 60
2000 PRINT"TIEDE"
2010 GOTO 60
3000 PRINT"CHAUD"
3010 GOTO 60
4000 PRINT"TRES CHAUD"
4010 GOTO 60
5000 PRINT"BRULANT"
5010 GOTO 60
6000 CLS
6010 PRINT"VOUS AVEZ TROUVE EN ";S;" COU
PS"
```

```
DONNEZ X :? 1
DONNEZ Y :? 3
VOUS ETES COMPLETEMENT GELE
DONNEZ X :? 1
DONNEZ Y :? 6
VOUS ETES COMPLETEMENT GELE
DONNEZ X :? 4
DONNEZ Y :? 1
VOUS ETES COMPLETEMENT GELE
DONNEZ X :? 6
DONNEZ Y :? 2
VOUS ETES TIEDE
DONNEZ X :? 5
DONNEZ Y :? 2
VOUS ETES CHAUD
```

Les allumettes

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : contre le MO6.
Difficulté du jeu : moyenne.

LE JEU :
Le joueur devra en début de partie choisir une certaine quantité d'allumettes, ainsi que la prise maximum autorisée. Il décide ensuite s'il commence, ou s'il laisse commencer le MO6. Chacun prend à son tour autant d'allumettes qu'il le désire, dans la limite du nombre maximum autorisé. Celui qui prend la dernière allumette est considéré comme perdant.

LE PROGRAMME :
Le jeu du joueur commence en ligne 7Ø.
Celui du MO6 commence en ligne 11Ø.
La ligne 8Ø contrôle la légalité du coup du joueur.
La ligne 9Ø gère le changement de joueur.
Le coup du MO6 est calculé dans une boucle non formelle, de la ligne 12Ø à la ligne 15Ø.
Le résultat est affiché par la ligne 1Ø2Ø.
Si la partie est terminée, la ligne 1Ø3Ø le détecte et envoie en ligne 2ØØØ, pour le message de fin.

POUR JOUER :
On commence l'exécution par RUN, et, à l'affichage du message NOMBRE TOTAL, on entre le nombre d'allumettes, suivi de ENTREE. Puis on entre la prise maximum autorisée, suivie de ENTREE. Si le joueur veut commencer, il frappe ensuite Ø ; sinon il frappe 1, puis ENTREE.
Après chaque coup, le nombre d'allumettes restantes est affiché, et le MO6 affiche VOUS RETIREZ lorsque c'est au joueur de prendre.
Celui qui prend la dernière allumette a perdu.

EXTENSIONS POSSIBLES :
On pourra bien sûr envisager un affichage graphique couleur plus spectaculaire, imaginer de laisser le hasard choisir le nombre d'allumettes et la prise maximum, etc.

```
10 REM LES ALLUMETTES
20 CLS
30 INPUT"NOMBRE TOTAL :";N
40 INPUT"PRISE MAXIMUM :";I
50 INPUT"QUI COMMENCE (VOUS:0 / MOI:1) :
";D
60 IF D=1 THEN 110
70 INPUT"VOUS RETIREZ :";P
80 IF P<>INT(P) OR P<1 OR P>N THEN 70
90 D=1
100 GOTO 1000
110 S=I+D
120 T=(N-S)/(I+D)
130 IF T=INT(T) THEN 200
140 S=S-D
150 GOTO 120
200 P=ABS(S-D)
210 IF P=0 THEN P=D
220 D=0
1000 PRINT P
1010 N=N-P
1020 PRINT"IL RESTE ";N
1030 IF N=0 THEN 2000
1040 GOTO 60
2000 CLS
2010 IF D=1 THEN PRINT"J'AI GAGNE":PLAY"
DO;LA;MI"
2020 IF D<>1 THEN PRINT"BRAVO":PLAY"O2;L
72;RE"
```

```
NOMBRE TOTAL :? 23
PRISE MAXIMUM :? 5
QUI COMMENCE (VOUS:0 / MOI:1) :? 1
 4
IL RESTE  19
VOUS RETIREZ :? 4
 4
IL RESTE  15
 2
IL RESTE  13
VOUS RETIREZ :? 5
 5
IL RESTE  8
 1
IL RESTE  7
VOUS RETIREZ :? 4
 4
IL RESTE  3
 2
IL RESTE  1
VOUS RETIREZ :? 1
 1
IL RESTE  0
```

Allumettes à plusieurs

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (plusieurs joueurs).
Difficulté du jeu : assez difficile.

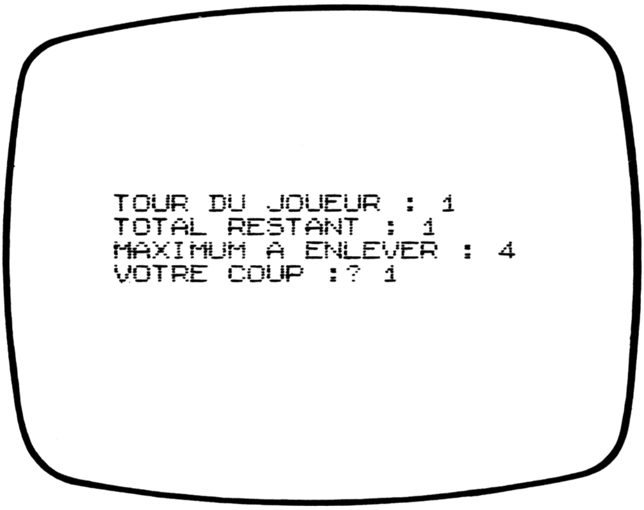
LE JEU :
Dans ce jeu pour plusieurs joueurs, le meneur de jeu devra, en début de partie, choisir un nombre d'allumettes, et indiquer au MO6 le nombre de joueurs.
A chaque coup, le joueur qui a le trait est autorisé à retirer du tas d'allumettes restantes jusqu'au double de ce que vient de retirer le précédent joueur ; celui qui retire la dernière allumette a gagné (ou perdu, si l'on modifie légèrement les règles).

LE PROGRAMME :
Les lignes 1Ø à 4Ø permettent les différentes entrées.
Les lignes 7Ø à 1ØØ réalisent les impressions.
La ligne 12Ø contrôle le coup de chaque joueur (par rapport au coup précédent).
La ligne 18Ø gère les tours (remise à 1 du numéro du joueur).
La ligne 16Ø réalise le test de fin, et la ligne 21Ø affiche le numéro du joueur gagnant (avec la règle prévue initialement).

POUR JOUER :
Après avoir lancé le programme par RUN, on entre le nombre d'allumettes initial, suivi de ENTREE, puis le nombre de joueurs (et ENTREE).
Le premier joueur ne pourra enlever qu'une allumette, le deuxième pourra en enlever 1 ou 2 ; le suivant 1, 2 ou 1, 2, 3 ou 4 selon le coup précédent, et ainsi de suite ; le retrait maximum autorisé est à chaque fois affiché par le MO6.
Lorsque la dernière allumette a été prise, on obtient l'affichage LE JOUEUR... A GAGNE, et la partie est terminée.

EXTENSIONS POSSIBLES :
On peut convenir que celui qui prend la dernière allumette est éliminé du jeu, et recommencer la partie entre les joueurs restants qui seront ainsi éliminés les uns après les autres.
On peut aussi imaginer une représentation graphique des tas d'allumettes, tirer au hasard le nombre à enlever en début de partie, etc.

```
10 REM ALLUMETTES A PLUSIEURS
20 CLS
30 INPUT"NOMBRE TOTAL D'ALLUMETTES :";N
40 INPUT"NOMBRE DE JOUEURS :";J
50 F=1
60 K=1
70 CLS
80 PRINT"TOUR DU JOUEUR :";K
90 PRINT"TOTAL RESTANT :";N
100 PRINT"MAXIMUM A ENLEVER :";F
110 INPUT"VOTRE COUP :";C
120 IF C=0 OR C>F THEN 110
130 N=N-C
140 IF N<0 THEN N=N+C:GOTO 110
150 F=C+C
160 IF N=0 THEN 200
170 K=K+1
180 IF K>J THEN K=1
190 GOTO 70
200 CLS
210 PRINT"LE JOUEUR NUMERO ";K;" A GAGNE
"
```



Calendrier perpétuel

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : gadget publicitaire.
Difficulté du jeu : aucune.

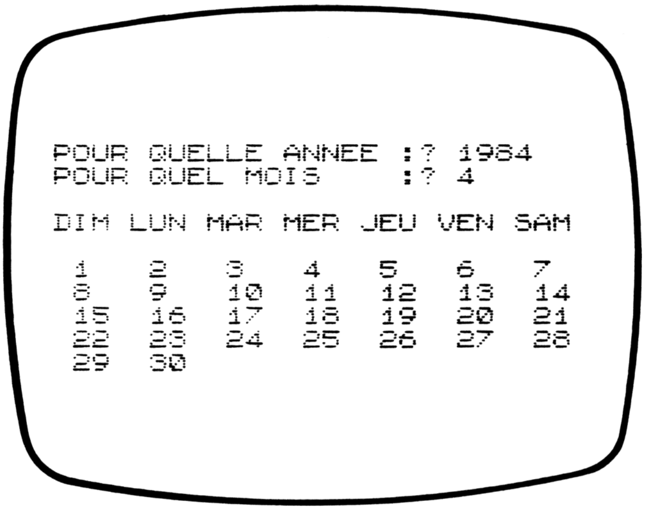
LE JEU :
Ce programme vous permettra de confectionner tous les ans votre calendrier ; vous pourrez ainsi vous en servir pour envoyer à vos amis un calendrier du mois de leur naissance, ou pour leur souhaiter une bonne fête, etc.

LE PROGRAMME :
Les lignes 3Ø et 4Ø vous permettront d'entrer les mois et l'année.
Les lignes 5Ø, 6Ø et 7Ø calculent le numéro du premier jour du mois.
La ligne 8Ø calcule le nombre de jours du mois (il n'est pas tenu compte ici des 29 février, pour simplifier).
La boucle A (12Ø-15Ø) réalise l'impression du calendrier proprement dit.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on devra introduire l'année, sous la forme, par exemple 1983, suivie de ENTREE.
On obtient alors la question MOIS, à laquelle il convient de répondre par le numéro du mois, suivant l'ordre usuel (1 pour JANVIER, 2 pour FEVRIER, etc.).
On fait ensuite ENTREE, et le calendrier est alors affiché sur l'écran.

EXTENSIONS POSSIBLES :
On pourra sans difficulté prévoir l'affichage des 29 février, prévoir une option permettant l'affichage pour une année complète, prévoir l'introduction du nom des mois en clair, etc.

```
10 REM CALENDRIER PERPETUEL
20 CLS
30 INPUT"POUR QUELLE ANNEE :";A
40 INPUT"POUR QUEL MOIS    :";M
50 D=INT(365.25*(A+(M<3)))+1
60 D=D+INT(30.6*(M+1-(M<3)*12))-INT((INT
((A+(M<3))/100)-7)*.75)
70 D=D-7*INT(D/7)
80 M=31+(M=4 OR M=6 OR M=9 OR M=11)+(M=2
)*3
90 PRINT
100 PRINT"DIM LUN MAR MER JEU VEN SAM"
110 PRINT
120 FOR A=1 TO M
130 PRINT TAB(4*(D+A-1));A;
140 IF D+A-1=6 THEN PRINT:D=D-7
150 NEXT A
```



Bataille navale

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

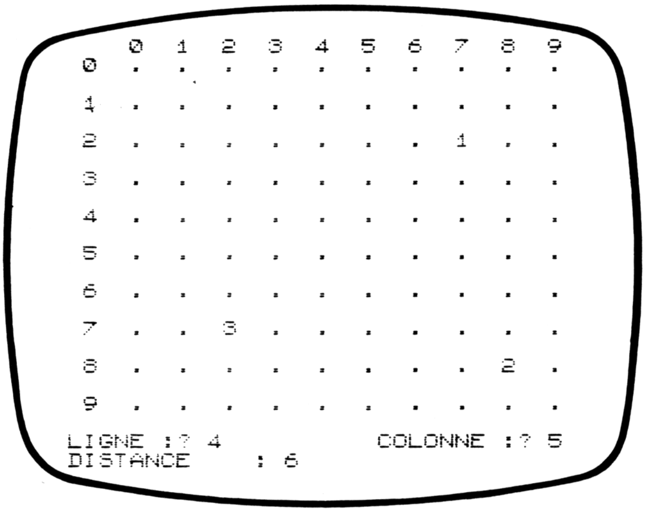
LE JEU :
Une bataille navale a priori assez simple, avec un seul navire caché au hasard dans une grille de 1Ø cases sur 1Ø cases.
A chaque coup, on devra tirer en donnant les coordonnées H (horizontale) et V (verticale) de la case visée, et le MO6 répond en affichant la distance séparant le navire invisible de la case visée. Mais, si l'on tire trop près, le navire risque de se recacher au hasard.

LE PROGRAMME :
La boucle I (4Ø-7Ø) dessine le terrain de jeu.
Les lignes 9Ø-1ØØ définissent l'emplacement du navire.
Les coordonnées du tir sont entrées aux lignes 11Ø et 12Ø, et la ligne 13Ø affiche le coup sur l'écran. La distance est calculée en ligne 14Ø, et la ligne 15Ø regarde si l'on a ou non fait mouche.
La ligne 17Ø intervient pour déplacer le navire si les coups sont rapprochés.
La ligne 18Ø permet l'affichage de la distance.

POUR JOUER :
Le terrain de jeu est affiché après avoir frappé RUN. A l'affichage de LIGNE, on répond en donnant le numéro de la ligne horizontale sur laquelle on désire tirer. On tape ensuite ENTREE et, à la question COLONNE, on répond en donnant le numéro de la colonne verticale suivi de ENTREE.
Si on a tiré trop près, on obtient l'affichage PFFFF... qui signifie que le navire ennemi a changé d'emplacement.
Si on a tiré sur la bonne case, on obtient l'affichage du nombre total de tirs effectués pour gagner.

EXTENSIONS POSSIBLES :
Elles sont innombrables, puisqu'aussi bien la bataille navale est l'un des jeux les plus répandus qui soit. On peut introduire deux joueurs, jouer contre le MO6, mettre plusieurs navires, des sous-marins, etc.

```
10 REM BATAILLE NAVALE
15 INPUT"DONNEZ UN ENTIER :";F
17 FOR I=1 TO F:Y=RND:NEXT I
20 CLS
30 PRINT"    0  1  2  3  4  5  6  7  8
9"
40 FOR I=0 TO 9
50 PRINT I;" .  .  .  .  .  .  .  .  .
."
60 PRINT
70 NEXT I
80 S=1
90 X=INT(RND*10)
100 Y=INT(RND*10)
110 LOCATE 0,21:INPUT"LIGNE :";H
120 LOCATE 20,21:INPUT"COLONNE :";V
130 LOCATE 3*V+3,2*H+1:COLOR 1,6:PRINT S
:COLOR 4,6
140 D=INT(SQR((X-H)*(X-H)+(Y-V)*(Y-V)))
150 IF D=0 THEN 200
160 S=S+1
170 IF D=1 THEN LOCATE 0,22:PRINT"PFFF..
.                    ":GOTO 90
180 LOCATE 0,22:PRINT"DISTANCE    :";D;"
      "
190 GOTO 110
200 CLS
210 PRINT"TROUVE EN ";S;" COUPS"
```



Golf

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

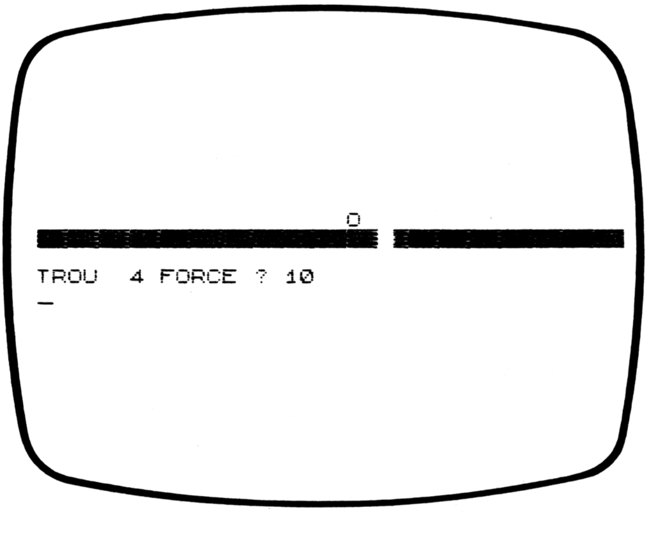
LE JEU :
Il s'agit d'essayer de réaliser, avec le moins de coups possible, un parcours de golf de 18 trous. Pour cela, il vous faut frapper la balle avec une force plus ou moins grande, jusqu'à obtenir sa chute dans le trou.
Mais attention, il peut y avoir un certain aléa sur chaque coup, et vous serez amené souvent à revenir en arrière (il faudra alors faire précéder l'indication de la force par un signe −).

LE PROGRAMME :
La ligne 7Ø permet de définir la position du trou, et la ligne 8Ø dessine le terrain de jeu.
Le déplacement à réaliser est calculé en ligne 12Ø.
La boucle I (14Ø-19Ø) réalise le déplacement de la balle.
La ligne 2ØØ regarde si le coup est, ou non, gagnant ; s'il l'est, on recommence (ligne 25Ø) jusqu'à ce que l'on ait réalisé 18 trous (boucle S (4Ø-25Ø) ; fin de boucle en 27Ø, avec affichage de fin).

POUR JOUER :
On obtient l'affichage du terrain en frappant RUN ainsi que le dessin de la balle sur sa position de départ, et le texte : TROU 1. FORCE ; on doit alors répondre en donnant la force désirée (un nombre entre 1 et 3ØØ...).
Dès que l'on aura tapé ENTREE, la balle recommence à se déplacer ; elle peut alors soit s'arrêter avant le trou (auquel cas on recommence), soit tomber dedans (on passe alors au trou suivant), soit le dépasser : on devra alors frapper avec une force négative.
Le jeu se termine avec l'affichage 18 TROUS EN ... COUPS.

EXTENSIONS POSSIBLES :
Il sera possible d'étendre le jeu pour faire participer plusieurs joueurs, d'améliorer la qualité graphique, d'utiliser une représentation spatiale du terrain (se référer à certains jeux d'arcade), d'avoir un terrain plus accidenté, des obstacles, de changer de club...

```
10 REM GOLF
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 FOR S=1 TO 18
50 CLS
60 X=1
70 T=INT(29*RND+9)
80 LOCATE 0,6:FOR I=1 TO 40:PRINT CHR$(1
27);:NEXT I
90 LOCATE T,6:PRINT" ":LOCATE X,5:PRINT"
O"
100 LOCATE 15,8:PRINT"                "
110 LOCATE 0,8:PRINT"TROU ";S;:INPUT"FOR
CE ";R
120 F=INT(R*0.16-(ABS(R)*(SQR(RND)+0.1)^
2)/1000)
130 IF F=0 THEN 100
140 FOR I=SGN(F) TO F STEP SGN(F)
150 X=X+SGN(F)
160 IF X<39 AND X>=1 THEN LOCATE X-SGN(F
),5:PRINT" ":LOCATE X,5:PRINT"O"
170 IF X>=38 THEN X=38
180 IF X<=1 THEN X=1
190 NEXT I
200 IF X=T THEN 230
210 E=E+1
220 GOTO 80
230 LOCATE X,5:PRINT" ":LOCATE X,6:PRINT
"O"
240 FOR P=1 TO 500:NEXT P
250 NEXT S
260 CLS
270 PRINT"18 TROUS EN ";E;" COUPS"
```



Solfège

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : facile.

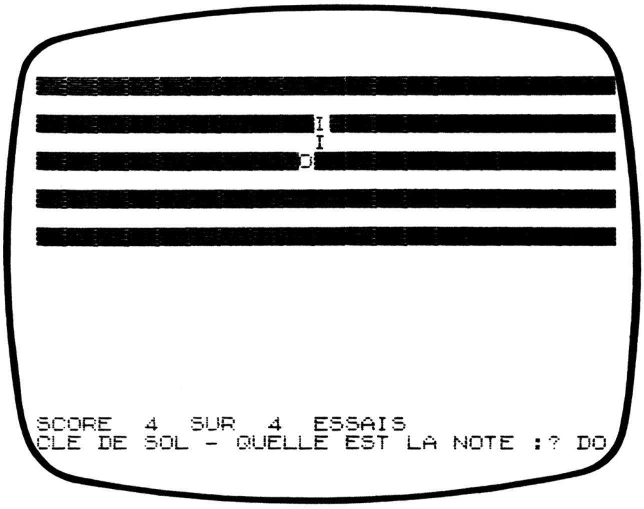
LE JEU :
Ce programme vous propose une initiation élémentaire au solfège, en vous permettant d'apprendre à reconnaître facilement les différentes notes, qui apparaissent les unes après les autres sur une portée ; après l'affichage de la note, vous devez entrer son nom en clair et votre score sera affiché en permanence.
Le programme est initialement prévu pour la clé de sol, mais il sera très facilement adaptable à toute autre clé.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine une portée.
La note est choisie en ligne 8Ø et affichée en lignes 9Ø-1ØØ-11Ø.
La ligne 12Ø affiche le score.
La ligne 13Ø affiche le nom de la clé (ici clé de sol) et vous interroge.
La ligne 14Ø s'occupe de la fin de la partie.
Les lignes 15Ø et 16Ø effectuent le décodage de la note entrée.
La ligne 19Ø contrôle le résultat.

POUR JOUER :
Après avoir lancé le programme par RUN, la portée sera affichée, ainsi que la première note.
Vous répondrez à la question NOTE en donnant le nom en clair de la note (do, ré, mi, fa, sol, la ou si), suivi de ENTREE.
Le score vous indiquera si c'est exact ou non, et une nouvelle note sera affichée.
Pour arrêter, il suffira de répondre Ø suivi de ENTREE, à la question NOTE.

EXTENSIONS POSSIBLES :
On pourra tout d'abord facilement modifier le programme pour lui permettre de vous faire travailler les différentes clés. On pourra aussi améliorer le graphisme, introduire la couleur, le son (lorsque l'on a répondu juste, on entendra la note, par exemple). On pourra aussi réaliser suivant le même principe un jeu vous permettant de reconnaître des accords, etc.

```
10 REM SOLFEGE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
35 S=0:P=0
40 CLS
50 FOR I=1 TO 5
60 LOCATE 0,2*I+1:FOR J=1 TO 40:PRINTCHR
$(127);:NEXT J
70 NEXT I
80 X=INT(RND*12)
90 LOCATE 18,X:PRINT"I"
100 LOCATE 18,X+1:PRINT"I"
110 LOCATE 17,X+2:PRINT"O"
120 LOCATE 0,21:PRINT"SCORE ";S;" SUR ";
P;" ESSAIS"
130 INPUT"CLE DE SOL - QUELLE EST LA NOT
E :";R$
140 IF R$="0" THEN END
150 Y=(1 AND R$="FA")+(2 AND R$="MI")+(3
 AND R$="RE")+(4 AND R$="DO")
160 Y=Y+(5 AND R$="SI")+(6 AND R$="LA")
170 P=P+1
180 IF R$="SOL" THEN R$="SO"
190 IF Y=X OR Y=X-7 THEN S=S+1:PLAY R$
200 GOTO 40
```



Le jeu du 21

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu contre le MO6.
Difficulté du jeu : facile.

LE JEU :
On devra entrer une mise au début du jeu, et on pourra lancer les dés en tapant 1 ; à chaque lancer, on obtiendra un nouveau total, qui augmentera peu à peu, et se rapprochera de 21. On devra essayer de se rapprocher le plus possible de 21, sans dépasser ce nombre. Lorsque l'on décidera de s'arrêter, il suffira de taper Ø pour laisser jouer le MO6 qui essaiera alors d'avoir un meilleur score.
A la fin de chaque partie, le gain sera affiché ainsi que le total des gains (ou des pertes).

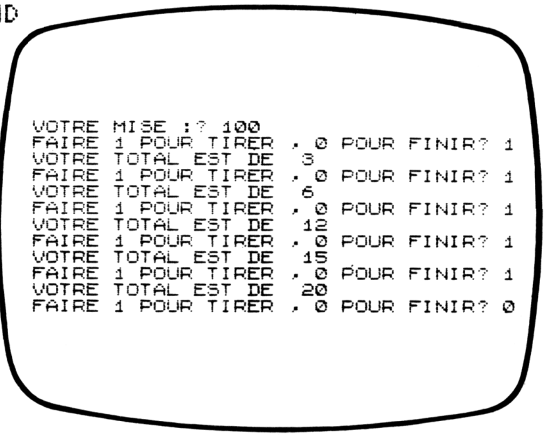
LE PROGRAMME :
Les lignes 8Ø et 9Ø réalisent le contrôle de celui qui doit jouer (joueur ou MO6).
Le sous-programme (1ØØØ-1Ø1Ø) lance les dés.
Les lignes 11Ø-12Ø calculent le total fait par le joueur et l'affichent.
Le total du MO6 est calculé en ligne 19Ø et affiché par la ligne 2ØØ.
La ligne 22Ø contrôle la fin du jeu pour le MO6.
Les lignes 26Ø-27Ø affichent les résultats, et la ligne 3ØØ cherche quel est le gagnant.
Le gain (ou la perte) sera affiché par la ligne 32Ø, et le total par la ligne 35Ø.
La ligne 38Ø contrôle la fin de partie.

POUR JOUER :
Taper RUN pour lancer le jeu.
A la question VOTRE MISE, entrer la mise initiale, suivie de ENTREE, et taper 1 pour lancer le dé. Le premier tirage est affiché sous la forme VOUS AVEZ... et on devra refaire 1 suivi de ENTREE, tant que l'on désire recommencer à tirer.
Dès que l'on a terminé, on tape Ø et ENTREE pour laisser jouer le MO6 (qui tient ici le rôle de la banque).
Le gain (ou la perte, s'il y a le signe −) sera affiché en fin de jeu, ainsi que le total des gains (ou pertes) réalisé depuis le début. Si l'on désire arrêter de jouer, on devra taper N en réponse à la question ON CONTINUE (O/N).

EXTENSIONS POSSIBLES :

Ce jeu sera évidemment plus intéressant s'il est pratiqué à plusieurs, et il sera aisé de modifier le programme en conséquence. On pourra aussi dessiner le dé, ou mieux, utiliser des cartes, ce qui permettra alors de faire un jeu de Black Jack.

```
10 REM JEU DU 21
20 G=0
30 INPUT"DONNEZ UN ENTIER :";F
40 FOR I=1 TO F:Y=RND:NEXT I
50 CLS
60 T=0
70 INPUT"VOTRE MISE :";M
80 INPUT"FAIRE 1 POUR TIRER , 0 POUR FIN
IR";R
90 IF R=0 THEN 150
100 GOSUB 1000
110 T=T+X
120 PRINT"VOTRE TOTAL EST DE ";T
130 GOTO 80
150 CLS
160 PRINT"LA BANQUE JOUE"
170 S=0
180 GOSUB 1000
190 S=S+X
200 PRINT"TOTAL DE LA BANQUE ";S
210 FOR I=1 TO 500:NEXT I
220 IF S>T OR S>17 THEN 250
230 GOTO 180
250 CLS
260 PRINT"VOUS AVEZ ";T
270 PRINT
280 PRINT"LA BANQUE A ";S
290 PRINT
300 IF S>21 OR (S<T AND T<=21) THEN 320
310 M=-M
320 PRINT"GAIN ";M
330 PRINT
340 G=G+M
350 PRINT"VOTRE TOTAL ACTUEL ";G
360 PRINT
370 INPUT"ON CONTINUE (O/N) :";R$
380 IF R$="N" THEN END
390 GOTO 50
1000 X=INT(RND*6+1)
1010 RETURN
```



Pair-impair

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

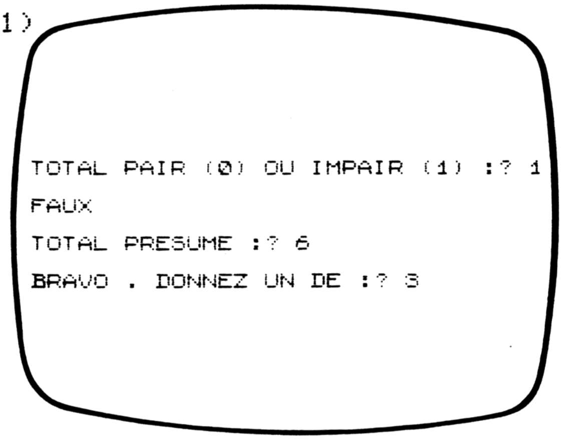
LE JEU :
C'est un jeu de hasard, pour un ou plusieurs joueurs. Le MO6 lance simultanément deux dés, non visibles et pose trois questions : quelle est la parité du total, quel est le total, et enfin donner la valeur de l'un des dés. On peut ensuite soit poursuivre le jeu, soit l'interrompre à tout moment.

LE PROGRAMME :
Les dés sont tirés par un sous-programme placé aux lignes 1000 et 1010.
La ligne 80 tient le compte du nombre de parties, et les lignes 180, 260 et 290 comptabilisent les points marqués.
La parité du total est déterminée par les lignes 130 et 140 (la ligne 140 compare avec la réponse du joueur).
La ligne 400 et les suivantes réalisent l'affichage de fin de partie : total, nombre de points, nombre de parties jouées.
La ligne 440 provoque l'arrêt si l'on répond N à la question ON CONTINUE.

POUR JOUER :
On lance le programme par l'instruction RUN.
Il suffit ensuite de répondre aux diverses questions posées, en suivant scrupuleusement les indications données et en tapant ENTREE après chaque réponse.
Lorsque l'on a terminé une partie, on obtient l'affichage du total actuel et du nombre de parties sous la forme de, par exemple, 6 SUR 10.
On obtient alors l'affichage de la question ON CONTINUE (O/N) et l'on devra répondre N suivi de ENTREE pour arrêter.

EXTENSIONS POSSIBLES :
Si l'on désire jouer à plusieurs, il sera intéressant de prévoir la possibilité de miser, et d'ajouter une gestion du score réalisé par chacun des joueurs.
On pourra aussi prévoir une amélioration sur le plan de la présentation (utilisation de la couleur, etc.).

```
10 REM PAIR-IMPAIR
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 T=0:R=0
50 GOSUB 1000
60 A=X
70 GOSUB 1000
80 R=R+1
90 S=A+X
100 CLS
110 INPUT"TOTAL PAIR (0) OU IMPAIR (1) :
";N
120 PRINT
130 P=S/2-INT(S/2)
140 IF N=P OR N-0.5=P THEN 170
150 PRINT"FAUX"
160 GOTO 190
170 PRINT"EXACT"
180 T=T+1
190 PRINT
200 INPUT"TOTAL PRESUME :";N
210 IF N=S THEN 250
220 PRINT"PERDU"
230 FOR I=1 TO 900:NEXT I
240 GOTO 400
250 PRINT
260 T=T+2
270 INPUT"BRAVO . DONNEZ UN DE :";N
280 IF N<>A AND N<>S-A THEN 220
290 T=T+3
300 PRINT
310 PRINT"BRAVO":GOTO 230
400 CLS
410 PRINT"VOUS AVEZ UN TOTAL DE ";T;" SU
R ";R
420 PRINT
430 INPUT"ON CONTINUE (O/N) :";R$
440 IF R$="N" THEN END
450 GOTO 50
1000 X=INT(RND*6+1)
1010 RETURN
```



Devine un chiffre

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
C'est un jeu de hasard, de conception très simple, mais néanmoins assez passionnant : le joueur introduit sa mise, et parie sur un chiffre, de 1 à 6 ; trois dés sont alors lancés ; si le chiffre joué apparaît deux fois il double sa mise ; s'il apparaît trois fois, il triple sa mise.

LE PROGRAMME :
Le tirage des dés est réalisé par un sous-programme placé en lignes 1ØØØ et 1Ø1Ø.
L'affichage est réalisé par les lignes 15Ø et 16Ø.
Les lignes 18Ø et 21Ø calculent le gain dans les diverses configurations possibles.
La totalisation est effectuée en ligne 23Ø pour les gains, et en ligne 51Ø pour les pertes.
La variable R$, testée en ligne 55Ø, détermine la poursuite éventuelle du jeu.

POUR JOUER :
Le lancement du programme par RUN provoque un premier affichage VOTRE MISE ; on répond par un nombre suivi de ENTREE. On obtient alors un deuxième affichage DONNEZ UN CHIFFRE DE 1 A 6 ; on entre un chiffre suivi de ENTREE ; les 3 dés, repérés par les lettres A, B, C s'affichent alors, ainsi qu'un message indiquant votre gain ou votre perte. Puis une indication sur votre total actuel apparaît, ainsi que la question ON CONTINUE (O/N) ; taper O pour continuer, N pour arrêter (suivi de ENTREE).

EXTENSIONS POSSIBLES :
On pourra facilement introduire des contrôles (mise minimum, coup proposé de 1 à 6, etc.). On pourra aussi dessiner des dés avec un graphisme plus élaboré, introduire la possibilité de jouer à plusieurs, faire même participer le MO6.

```
10 REM DEVINE UN CHIFFRE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 G=0
50 CLS
60 INPUT"VOTRE MISE :";N
70 PRINT
80 INPUT"DONNEZ UN CHIFFRE DE 1 A 6 :";D
90 GOSUB 1000
100 A=X
110 GOSUB 1000
120 B=X
130 GOSUB 1000
140 C=X
150 PRINT:PRINT:PRINT"     A              B
             C"
160 PRINT:PRINT"    ";A;"              ";B;"
          ";C
170 T=0
180 IF D=A THEN T=T+N
190 IF D=B THEN T=T+N
200 IF D=C THEN T=T+N
210 IF T=0 THEN 500
220 PRINT:PRINT:PRINT"VOUS GAGNEZ ";T
230 G=G+T
240 GOTO 520
500 PRINT:PRINT:PRINT"PERDU              "
510 G=G-N
520 PRINT:PRINT:PRINT"TOTAL ACTUEL :";G
530 PRINT
540 INPUT"ON CONTINUE (O/N) :";R$
550 IF R$="N" THEN END
560 GOTO 50
1000 X=INT(RND*6+1)
1010 RETURN
```

```
VOTRE MISE :? 100

DONNEZ UN CHIFFRE DE 1 A 6 :? 3


    A             B             C

    6             6             3


VOUS GAGNEZ  100


TOTAL ACTUEL : 0

ON CONTINUE (O/N) :? N
```

Le jeu de Wytthof

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

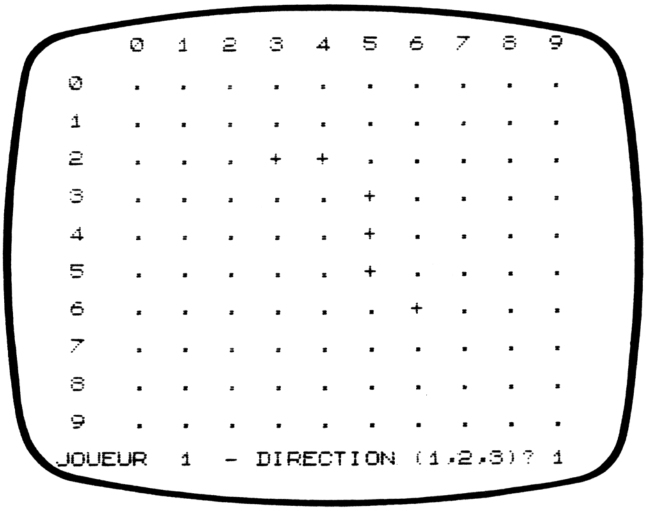
LE JEU :
On dispose d'une grille carrée de 10 cases sur 10 cases. Un point apparaît, en dehors de la diagonale et les joueurs devront le déplacer à chaque tour, suivant seulement trois directions : droite, bas et diagonale, bas et droite. Le premier joueur qui arrive en (9,9) sera considéré comme gagnant, ou perdant, selon la convention faite au début.

LE PROGRAMME :
Les lignes 40, 50 et 60 permettent de définir une position de départ, en dehors de la diagonale.
La ligne 80 et la boucle I (90-110) dessinent le terrain de jeu.
La ligne 130 imprime la position du point mobile.
Le test de fin est effectué en ligne 140.
Le changement de joueur est assuré par la ligne 180.
Les directions de déplacement disponibles, 1, 2 et 3, sont décodées par trois sous-programmes qui commencent aux lignes 1000, 2000 et 3000.

POUR JOUER :
Taper RUN pour lancer l'exécution du programme. Le terrain de jeu s'affiche, ainsi que la position de départ.
Puis on obtient l'affichage JOUEUR 1 et en dessous la question DIRECTION (1, 2, 3), à laquelle on répond en tapant l'un de ces trois nombres, suivi de ENTREE.
C'est ensuite au tour du joueur 2.
Le jeu se termine lorsque l'un des joueurs arrive en case (9, 9).

EXTENSIONS POSSIBLES :
On peut tout d'abord améliorer les contrôles (la sortie du terrain n'est pas interdite, par exemple), améliorer ensuite l'affichage (graphisme, couleur), étendre le jeu à plusieurs joueurs avec un terrain agrandi, introduire un monstre qui va essayer de capturer un des joueurs, prévoir un programme qui permettrait au MO6 de jouer, etc.

```
10 REM LE JEU DE WYTTHOF
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 X=3*INT(RND*5)+5
50 Y=INT(RND*5)
60 IF 3*Y+5=X THEN 50
70 CLS
80 PRINT"      0  1  2  3  4  5  6  7  8
 9"
90 FOR I=0 TO 9
100 PRINT:PRINT I;"  .  .  .  .  .  .  .
  .  .  ."
110 NEXT I
120 H=1
130 LOCATE X,2*Y+2:PRINT"+"
140 IF X=32 AND Y=9 THEN END
150 LOCATE 0,22:PRINT"JOUEUR ";H;:INPUT"
 - DIRECTION (1,2,3)";D$
160 D=VAL(D$)
170 ON D GOSUB 1000,2000,3000
180 H=2-ABS(H-1)
190 GOTO 130
1000 X=X+3
1010 RETURN
2000 GOSUB 1000
3000 Y=Y+1
3010 RETURN
```



Rebonds

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu d'adresse.
Difficulté du jeu : moyenne.

LE JEU :
Il utilise l'idée de la simulation d'un billard à bandes.
De part et d'autre de l'écran se trouvent la balle et le but visé : une série de chiffres et de lettres, en haut de l'écran, permettent de choisir l'endroit où l'on désire rebondir.
La balle va alors partir vers le haut de l'écran (vers le bas si l'on introduit un signe moins), y rebondir, revenir éventuellement frapper le bas, rebondir, et ainsi de suite jusqu'à atteindre la ligne de but ; si l'on est suffisamment près de la cible on a gagné et le nombre de rebonds sera alors affiché.

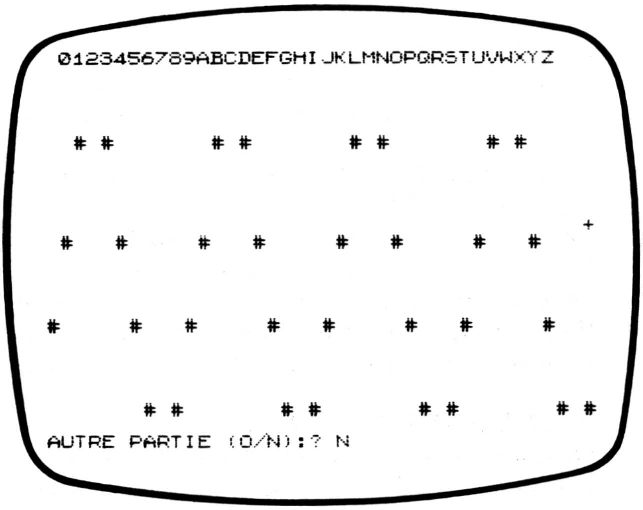
LE PROGRAMME :
La ligne 5Ø imprime les chiffres et les lettres en haut de l'écran.
Les lignes 6Ø et 7Ø définissent les positions initiales de la balle et de la cible.
Les lignes 8Ø et 9Ø effectuent leur impression.
Le symbole correspondant à la direction choisie est entré en ligne 1ØØ.
La boucle J (13Ø-17Ø) réalise le déplacement de la balle.
La ligne 16Ø imprime la nouvelle position.
Le changement de direction, lors de la rencontre d'un bord, est effectué par la ligne 15Ø, par renvoi dans une partie du programme indépendante, située aux lignes 22Ø à 24Ø.
La ligne 18Ø effectue l'impression de fin de partie.
Le nombre de rebonds est compté en ligne 23Ø.

POUR JOUER :
On lance le programme par RUN.
Le terrain de jeu est affiché, ainsi que les positions initiales de la balle et de la cible. On choisit l'endroit où l'on désire faire rebondir la balle (de 1 à Z), et on l'entre suivi de ENTREE.
La trajectoire est alors affichée, suivie s'il y a lieu de la mention BRAVO ; X REBONDS.
On peut lancer une autre partie en tapant une touche quelconque, et s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

Il sera très intéressant, en utilisant un graphisme plus fin, de dessiner un véritable billard, avec son tapis vert, et d'indiquer la direction par un segment, symbolisant la queue de billard, se déplaçant autour de la bille. On peut aussi ajouter des trous, mettre plusieurs billes, sonoriser les rebonds, ou les heurts avec d'autres balles, les comptabiliser pour faire une partie à plusieurs joueurs, etc.

```
10 REM REBONDS
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 PRINT" 0123456789ABCDEFGHIJKLMNOPQRST
UVWXYZ"
60 X=INT(RND*18+2)
70 Y=INT(RND*18+2)
80 LOCATE 0,X:PRINT"#"
90 LOCATE 39,Y:PRINT"+"
100 LOCATE 0,23:INPUT"VOTRE COUP :";R$
110 S=0
120 T=X
130 FOR J=1 TO 39
140 T=T-X/(ASC(R$)-ASC("0")+7*(R$>="A"))
150 IF T>=22 OR T<=0 THEN 220
160 LOCATE J,T:PRINT"#"
170 NEXT J
180 IF ABS(T-Y)<3 THEN PRINT"BRAVO :";S;
" REBONDS":PLAY"DO;SO;MI"
190 INPUT"AUTRE PARTIE (O/N):";R$
200 IF R$="N" THEN END
210 GOTO 40
220 X=-X
230 S=S+1
240 GOTO 170
```



Jackpot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (un joueur).
Difficulté du jeu : facile.

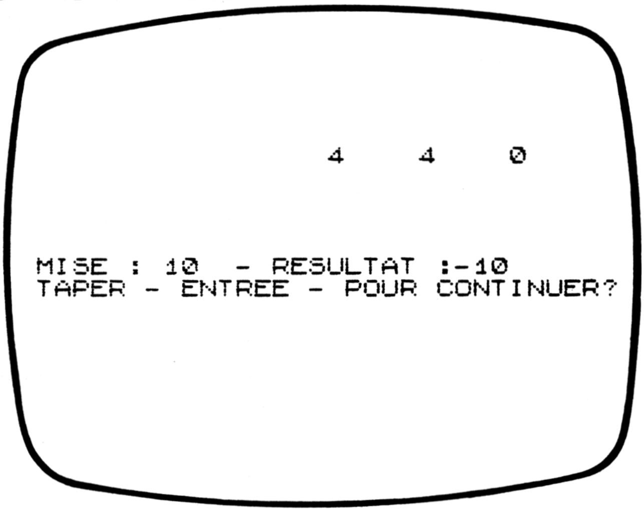
LE JEU :
C'est une variante simplifiée d'un jeu de hasard classique, dans lequel le joueur doit faire une mise. S'il obtient certaines combinaisons de symboles, il gagne (plus ou moins, selon sa mise et la combinaison obtenue).
Ici, les symboles sont remplacés par des chiffres, et le fait d'avoir trois chiffres identiques permet de gagner. Il existe de plus une combinaison particulière qui permet de remporter le Jackpot.

LE PROGRAMME :
La boucle I (6Ø-8Ø) permet le dessin de la machine, la ligne 9Ø affichant les emplacements (vides) des chiffres.
La mise sera entrée en ligne 1ØØ, et testée en ligne 11Ø : si on mise Ø, la partie est terminée (départ vers la ligne 2ØØØ).
Les lignes 12Ø, 14Ø et 16Ø permettent le tirage des 3 chiffres, en appelant un sous-programme qui se trouve en ligne 1ØØØ.
Le résultat est affiché par la ligne 17Ø, puis testé en ligne 18Ø, et en ligne 31Ø, si l'on a gagné.

POUR JOUER :
Lancer le programme par RUN ; à l'affichage de la question POT DE DEBUT, on répond en entrant un nombre suffisamment élevé, sur lequel vos mises seront décomptées.
Puis on tape ENTREE, et on entre une première mise (différente de Ø), suivie de ENTREE.
Les trois chiffres tirés sont alors affichés un bref instant, ainsi que la mise faite, et le résultat obtenu (gain ou perte).
On peut ensuite miser de nouveau, faire ENTREE et ainsi de suite.
Pour terminer, il suffit de répondre Ø à la question VOTRE MISE.
Le restant de votre pot initial sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra envisager d'améliorer sérieusement les contrôles, utiliser la couleur et des possibilités graphiques plus élaborées pour simuler davantage un véritable Jackpot.

```
10 REM JACKPOT
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 INPUT"POT DE DEBUT :";P
50 CLS
60 FOR I=1 TO 3
70 LOCATE 10,I+4:COLOR 3,1:PRINT "
             ":COLOR 0,6
80 NEXT I
90 LOCATE 14,6:PRINT"   ":LOCATE 19,6:PR
INT"   ":LOCATE 24,6:PRINT"   "
100 LOCATE 0,11:INPUT"VOTRE MISE :";M
110 IF M=0 THEN 2000
120 GOSUB 1000
130 A=X
140 GOSUB 1000
150 B=X
160 GOSUB 1000
170 LOCATE 15,6:PRINT A:LOCATE 20,6:PRIN
T B:LOCATE 25,6:PRINT X
180 IF A=B AND A=X THEN 300
190 N=-M
200 LOCATE 0,11:PRINT"MISE :";M;" - RESU
LTAT :";N;"    "
210 INPUT"TAPER - ENTREE - POUR CONTINUE
R";R$
220 P=P+N
230 GOTO 50
300 N=M*50
310 IF A=7 THEN N=M*500
320 GOTO 200
1000 X=INT(RND*9+1)
1010 RETURN
2000 CLS
2010 PRINT"VOUS AVEZ ";P
2020 PRINT"A BIENTOT"
```



Roulette

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

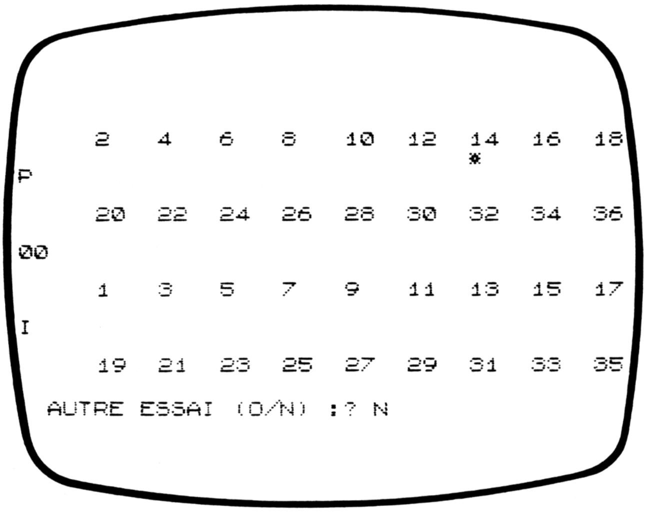
LE JEU :
Il est surtout spectaculaire par son aspect graphique : une boule, représentée par un carré noir, se déplace sur un terrain de jeu comportant des cases numérotées de 1 à 36, ainsi que trois cases supplémentaires, le ØØ, une case P (comme pair) et une case I (comme impair).
La boule parcourt au moins une fois chaque case, puis s'arrête au deuxième tour, on peut ensuite la relancer pour une autre partie.

LE PROGRAMME :
La boule K (5Ø-11Ø) dessine le terrain de jeu, auquel la ligne 12Ø rajoute les trois symboles P, ØØ et I.
La boucle K (13Ø-2ØØ) réalise le déplacement de la boule, l'arrêt étant contrôlé par les lignes 16Ø et 18Ø.
Le sous-programme 3ØØ-31Ø réalise une légère temporisation.
La ligne 21Ø tire la position d'arrêt.
La ligne 23Ø gère les essais suivants.

POUR JOUER :
Il suffit de faire RUN, et de regarder.
Une fois la boule arrêtée, il suffira de frapper une touche quelconque (sauf "N") pour la relancer. Taper N pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra essayer de se rapprocher davantage d'une véritable roulette, de différentes manières : mieux dessiner le terrain de jeu, introduire des couleurs, simuler le ralentissement de la boule, introduire différentes possibilités de pari, et assurer la gestion des gains correspondants, tenir les comptes de plusieurs joueurs, et annoncer les résultats sous une forme plus usuelle.

```
10 REM ROULETTE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 FOR K=1 TO 4
60 FOR I=1 TO 9
70 S=I+I
80 PRINT TAB(4*I)(S AND K=1)+(S+18 AND K
=2)+(S-1 AND K=3)+(S+17 AND K=4);
90 NEXT I
100 PRINT:PRINT:PRINT
110 NEXT K
120 LOCATE 0,2:PRINT"P":LOCATE 0,6:PRINT
"00":LOCATE 0,10:PRINT"I"
130 FOR K=1 TO 4
140 FOR I=1 TO 10
145 IF I=10 AND K<4 THEN LOCATE 0,4*K-1:
GOTO 160
150 LOCATE 4*I+1,4*K-3
160 X=X-1:BEEP
170 GOSUB 300
180 IF X=1 THEN PRINT"*":GOTO 230
190 NEXT I
200 NEXT K
210 X=INT(RND*38)
220 GOTO 120
230 LOCATE 0,20:INPUT"AUTRE ESSAI (O/N)
:";R$
240 IF R$="N" THEN END
250 GOTO 40
300 FOR J=1 TO 250:NEXT J
310 RETURN
```



Couloirs

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

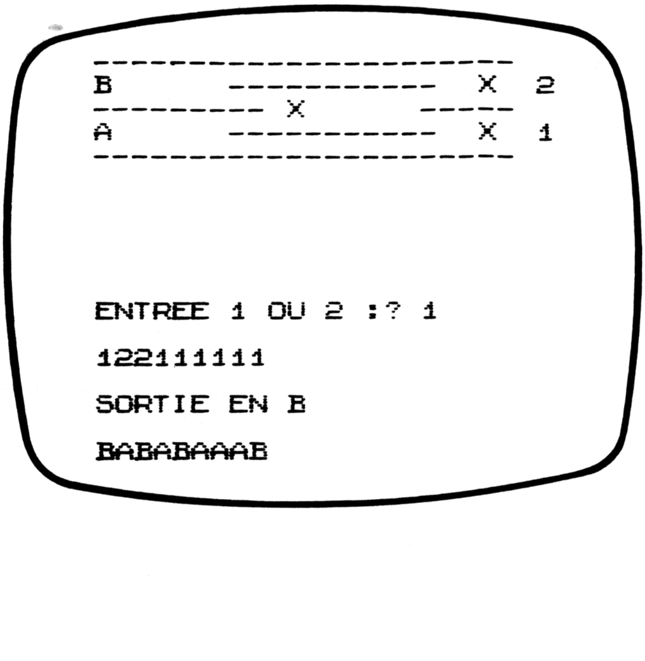
LE JEU :
Il fait appel à des notions de logique élémentaire. Le joueur peut se présenter à l'entrée, soit du couloir numéro 1, soit du couloir numéro 2. Dans ces deux entrées sont disposés des tourniquets qui basculent à votre passage et vous laissent passer, une fois vers la droite, et la fois suivante vers la gauche. De plus, un tourniquet analogue a été disposé dans le couloir central, pourrez-vous prévoir où (sortie A ou B) va s'effectuer votre prochaine sortie ?

LE PROGRAMME :
Les lignes 3Ø à 7Ø dessinent les couloirs (et les trois tourniquets).
La ligne 1ØØ attend votre choix.
Le décodage du chemin est fait par le MO6 en utilisant trois variables logiques I, J et K, représentant les états des trois tourniquets (lignes 13Ø à 27Ø).

POUR JOUER :
Après la frappe de RUN, le couloir est dessiné sur l'écran, les tourniquets étant visibles, et l'on devra taper 1 ou 2 en réponse à la question du MO6. Dans la version retenue ici, la réponse est alors immédiatement affichée, et le chemin suivi depuis le début du jeu, ainsi que les sorties correspondantes, sont affichés sur l'écran. Ceci présente l'avantage de permettre à plusieurs joueurs d'exercer leur sagacité en essayant de prévoir, pour une entrée donnée, ce que va répondre le MO6.

EXTENSIONS POSSIBLES :
Elles seront très nombreuses ; il est très facile d'imaginer un couloir plus complexe avec d'autres entrées et d'autres sorties, et d'autres tourniquets. On peut imaginer un affichage graphique des résultats en dessinant, un bref instant, l'état des tourniquets après la réponse du joueur. On peut aussi ajouter un test des réponses (jeu à plusieurs), décompter les points de chaque joueur, etc.).

```
10 REM COULOIRS
20 CLS
30 PRINT"----------------------"
40 PRINT"B       -----------  X  2"
50 PRINT"--------- X      -----"
60 PRINT"A       -----------  X  1"
70 PRINT"----------------------"
80 I=0:J=0:K=0
90 R$="":T$=""
100 LOCATE 0,13:INPUT"ENTREE 1 OU 2 :";C
$
110 IF C$="" OR C$>="3" OR LEN(C$)<>1 TH
EN 100
120 R$=R$+C$
130 PRINT:PRINTR$
140 IF C$="1" THEN 190
150 J=NOT(J)
160 IF J<>0 THEN 250
170 S$="B"
180 GOTO 220
190 I=NOT(I)
200 IF I<>0 THEN 250
210 S$="A"
220 T$=T$+S$
230 LOCATE 0,17:PRINT"SORTIE EN ";S$:PRI
NT:PRINT T$
240 GOTO 100
250 K=NOT(K)
260 IF K<>0 THEN 170
270 GOTO 210
```



Echiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : graphisme.
Difficulté du jeu : variable, selon utilisation.

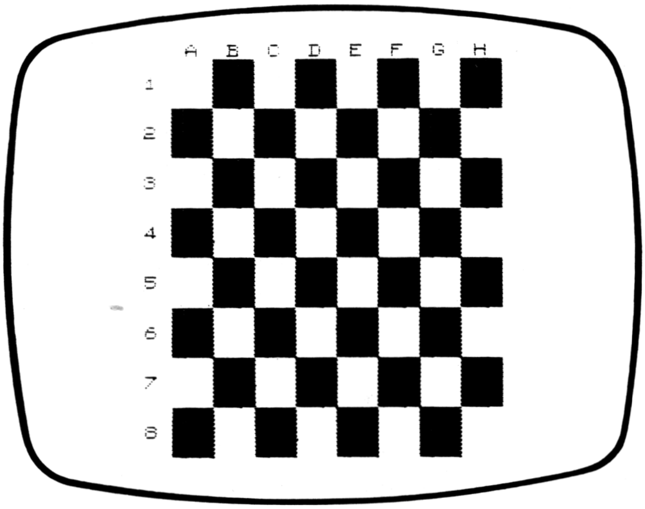
LE JEU :
Le programme proposé vient en appoint du programme Echecs ; il donne, en effet, la possibilité de réaliser un affichage graphique de bonne qualité pour des jeux utilisant un damier de 8 cases (on pourra facilement l'étendre à 10 cases sur 10 cases). Un repérage a été prévu, horizontalement, par des lettres (de A à H), et, verticalement, par des nombres (de 1 à 8).

LE PROGRAMME :
Les lignes 30, 40 et 50 définissent trois variables A$, B$ et C$ qui contiennent les éléments de dessin. La ligne 60 imprime les lettres horizontales et la première ligne du dessin.
La boucle I (70-130) permet l'impression, le motif à imprimer étant affiché ou non selon la valeur de (I + J)/2 ; la ligne 100 réalise l'impression.

POUR JOUER :
L'ordre RUN provoque l'affichage du damier sur l'écran.
Pour utiliser le programme, il conviendra de tenir compte du repérage des cases utilisables qui sont les centres des carrés noirs ou gris. Les cases sont obtenues en largeur par une tabulation de la forme 3*J, J variant de 1 à 8, et en hauteur par 3*I−1.

EXTENSIONS POSSIBLES :
On pourra utiliser ce programme pour toute sorte de jeux se déroulant sur un damier (échecs, dames anglaises, le loup et les agneaux, etc.).

```
10 REM ECHIQUIER
20 CLS
30 A$=CHR$(127)+CHR$(127)+CHR$(127)
40 B$=A$
50 C$=B$
60 PRINT"     A  B  C  D  E  F  G  H"
70 FOR I=1 TO 8
80 FOR J=1 TO 8
90 IF (I+J)/2=INT((I+J)/2) THEN 120
100 COLOR 3,1:LOCATE 3*J,3*I-2:PRINT A$:
PRINT I:LOCATE 3*J,3*I-1:PRINT B$:LOCATE
 3*J,3*I:PRINT C$;
120 NEXT J
130 NEXT I
140 LOCATE 0,0,0
```



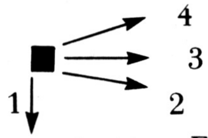
Le jeu de la fève

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Présentant une certaine analogie avec le jeu de WYTTHOF, cette version offre en fait 4 possibilités de déplacement ; la position de départ est ici fixée en (Ø, Ø), et pourrait éventuellement être modifiée.
Les directions de déplacement disponibles sont : droite, bas, diagonale bas et droite, diagonale haut et droite.
On joue en principe avec la règle : le premier arrivé en (9, 9) a perdu.

LE PROGRAMME :
La ligne 3Ø et la boucle I (4Ø-6Ø) dessinent le terrain de jeu.
La ligne 1ØØ réalise le test de fin de partie.
La ligne 13Ø permet d'exécuter l'un des sous-programmes de déplacement.
Les lignes 14Ø et 15Ø assurent le changement de joueur.

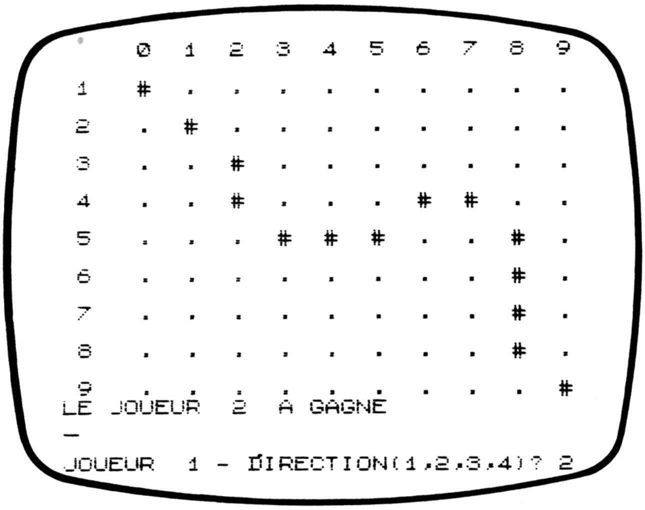
POUR JOUER :
On lance le programme par RUN, et le terrain de jeu s'affiche, ainsi que la question DIRECTION (1, 2, 3, 4) ; les 4 directions proposées correspondent aux déplacements suivants :



Le joueur 1 entre son déplacement, et tape ENTREE.
Le mouvement correspondant est réalisé, puis on obtient l'affichage JOUEUR 2 et DIRECTION.
Le joueur 2 procède de même, et ainsi de suite, jusqu'à ce que l'un des joueurs arrive en (9, 9) auquel cas le jeu se termine.

EXTENSIONS POSSIBLES :
On peut imaginer d'introduire en début de partie des cases pièges (chacun essayant alors de faire venir l'autre sur l'une de ces cases).
On peut aussi jouer avec plusieurs symboles, si possible de couleurs différentes, déplacés simultanément, envisager des terrains de jeu triangulaires, aux dimensions arbitraires, etc.

```
10 REM LE JEU DE LA FEVE
20 CLS
30 PRINT"      0  1  2  3  4  5  6  7  8
 9"
40 FOR I=1 TO 9
50 PRINT:PRINT I;"  .  .  .  .  .  .  .
 .  .  ."
60 NEXT I
70 X=5:Y=1
80 J=1
90 LOCATE X,2*Y:PRINT"#"
100 IF X=32 AND Y=9 THEN 200
110 LOCATE 0,22:PRINT"JOUEUR ";J;:INPUT"
- DIRECTION(1,2,3,4)";I$
120 I=VAL(I$):BEEP
130 ON I GOSUB 1000,2000,3000,4000
140 J=J+1
150 IF J=3 THEN 80
160 GOTO 90
200 PRINT"LE JOUEUR ";J;" A GAGNE
        ":PLAY"DO;SO;MI"
210 END
1000 Y=Y+1
1010 RETURN
2000 GOSUB 1000
3000 X=X+3
3010 RETURN
4000 GOSUB 3000
4010 Y=Y-1
4020 RETURN
```



Jeu de l'oie aléatoire

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard (deux joueurs).
Difficulté du jeu : facile.

LE JEU :
Le jeu reprend la très ancienne idée du jeu de l'oie, mais en y ajoutant la possibilité offerte par l'ordinateur de choisir au hasard les cases "à problème".
La règle du jeu est sans grand intérêt, puisque les joueurs n'interviennent à aucun moment dans le déroulement de la partie, qui est entièrement gérée par le MO6 ; il suffira d'appuyer sur une touche après chaque lancer de dé, et de regarder ce qui se passe.

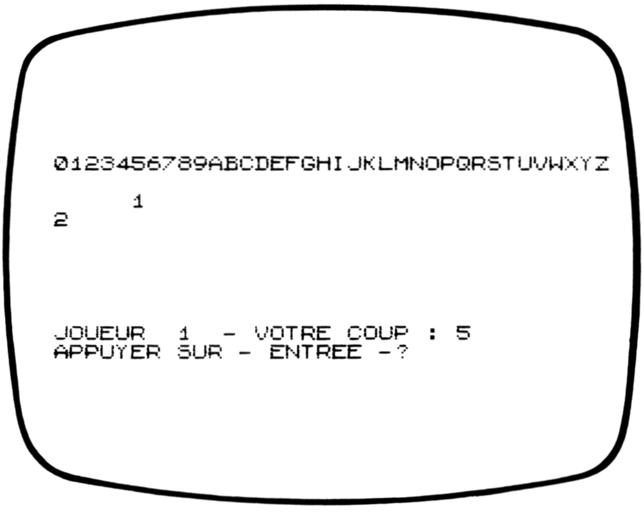
LE PROGRAMME :
La ligne 6Ø permet d'appeler le sous-programme 1ØØØ à 11ØØ qui réalise l'affichage du terrain de jeu, le contrôle de la position des joueurs et l'affichage de cette position. Cet affichage sera d'ailleurs refait en cours de partie, pour tenir compte du déplacement réellement effectué (qui n'est pas nécessairement celui indiqué par le dé).
La ligne 7Ø s'occupe du changement de joueur, et la ligne 8Ø effectue le lancer de dé.
La ligne 9Ø traite le cas où l'un des joueurs termine.
La ligne 13Ø tire un nombre aléatoire entre Ø et 3, qui servira s'il y a lieu pour déterminer le vrai déplacement du joueur, en envoyant dans l'un des sous-programmes de "chance".
Les lignes 1ØØØ et 1Ø1Ø font avancer les pions, et les lignes 1Ø2Ø à 1Ø8Ø redessinent à chaque affichage le terrain de jeu en entier.

POUR JOUER :
On lancera le jeu par RUN. Le terrain de jeu est alors dessiné, puis on voit apparaître l'affichage JOUEUR 1 :...
On devra alors appuyer sur une touche quelconque pour que le coup du joueur numéro 1 soit pris en compte ; ne pas s'inquiéter du temps de réaction, qui pourra sembler un peu long.
Le déplacement indiqué par les dés sera exécuté, puis au bout de quelque temps, un second déplacement sera éventuellement effectué : c'est le déplacement "CHANCE". On obtiendra alors l'affichage JOUEUR 2 :...
Le joueur 2 appuie à son tour sur une touche quelconque et ainsi de suite...
Le joueur gagnant sera signalé par un affichage BRAVO en fin de partie.
On peut aussi s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :

On peut agrandir le terrain de jeu, introduire davantage d'aléas, changer leur nature. On pourrait aussi choisir les cases “chance” en début de jeu, ajouter des affichages en conséquence, etc.

```
10 REM JEU DE L'OIE ALEATOIRE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 X=0:Y=0:J=1:I=X
60 GOSUB 1000
70 J=ABS(J-1)
80 I=INT(RND*6+1)
90 IF X=35 OR Y=35 THEN LOCATE 0,20:PRIN
T"BRAVO":PLAY"DO;SO;MI":END
100 BEEP:LOCATE 0,12:PRINT"JOUEUR ";J+1;
" - VOTRE COUP :";I
110 INPUT"APPUYER SUR - ENTREE -";R$
120 GOSUB 1000
130 I=INT(RND*4)
140 F=INT(RND*5+1)
150 ON F GOSUB 200,300,400,500,160
160 GOTO 60
200 I=I+3
210 RETURN
300 I=-I
310 RETURN
400 I=2*I
410 RETURN
500 IF J=0 THEN X=INT(RND*30):RETURN
510 Y=INT(RND*30):RETURN
1000 IF J=0 THEN X=X-I*(X+I<=35)
1010 Y=Y-J*I*(Y+I<=35)
1020 LOCATE 0,3:PRINT"0123456789ABCDEFGH
IJKLMNOPQRSTUVWXYZ"
1030 FOR I=0 TO 35
1040 LOCATE I,5:COLOR 3,1:PRINT" ";
1050 IF I=X THEN LOCATE I,5:PRINT"1";
1060 LOCATE I,6:COLOR 2,4:PRINT" ";
1070 IF I=Y THEN LOCATE I,6:PRINT"2";
1080 NEXT I
1090 FOR I=1 TO 600:NEXT I
1100 COLOR 0,6
1110 RETURN
```



Niveau 2

Nom du programme	*Thèmes et particularités*
Dé truqué	A utiliser, si vous êtes malhonnête, dans d'autres programmes.
Visions	Un jeu sur les mots, pour l'ordinateur.
421	Une simulation simplifiée, plusieurs joueurs.
L'électricien fou	Logique et difficile, pour un seul joueur.
La course au 20	Jeu de réflexion, pour deux joueurs ou plus.
Musique	Révisez vos notions de solfège, grâce à l'ordinateur.
Poker	Une base de départ, pour mettre au point votre programme.
Le jeu de saute-mouton	Un jeu de réflexion, pour deux joueurs.
Lettres	Un jeu sur les mots, assez simple.
Machinamot	Laissez l'ordinateur créer des vocables nouveaux.
Espion	Dites-lui votre amour en code.
Accrochez les wagons	Pour les plus jeunes ; très visuel.
Tirage d'un dé truqué	Révisez vos connaissances en statistiques.
Le jeu de Marienbad	Un jeu de logique pour deux joueurs, dans une version étendue.
Compose	Faire de la musique, c'est d'abord l'écrire...
Jour de la semaine	Profitez-en pour faire le point sur les anniversaires.
Le jeu de la vie	Un grand classique, en deux versions : manuelle et automatique.
Tour de cartes	Votre ordinateur a plus d'un tour dans son sac.
L'indésirable	Vous êtes propriétaire. Arriverez-vous à mettre la main sur ce locataire indésirable ?

Dimensionnement d'un tableau Recherche d'un élément dans un tableau et dans une chaîne L'instruction DATA

L'INSTRUCTION DIM OU INSTRUCTION DE DIMENSIONNEMENT D'UN TABLEAU

Les structures de tableaux sont très couramment utilisées en programmation, se révélant même indispensables dans un grand nombre de cas. Si vous voulez, par exemple, entrer vos recettes et dépenses pour tous les jours du mois (on supposera pour simplifier des mois de trente jours) vous pourriez définir 30 variables appelées PREMIER JOUR, DEUXIEME JOUR, etc. Il est plus facile, et plus rationnel d'utiliser une boucle I, I variant de 1 à 3Ø, et une variable R (I) pour les recettes, et D (I) pour les dépenses ; ce faisant, vous avez, sans le savoir, utilisé une structure de tableau à une dimension ; ceci ne pourra être admis par le MO6 que si vous l'avez auparavant prévenu que vous alliez utiliser un tel tableau, en donnant une instruction DIM R (3Ø) pour les recettes, et DIM D (3Ø) pour les dépenses.

On pourrait aussi imaginer de conserver dépenses et recettes dans un même tableau, en lui donnant cette fois deux dimensions : par exemple DIM RD (3Ø, 2) (ou DIM RD (2, 3Ø)). Ainsi, avec DIM RD (3Ø, 2) on pourra convenir que RD (I, 1) représente les recettes, et RD (I, 2) les dépenses. Des tableaux de plus de 2 dimensions sont également possibles.

On peut aussi concevoir des tableaux dont les éléments sont des chaînes de caractères, et non plus des nombres, avec 255 caractères au maximum.

Ainsi DIM NOM$ (12) réserve 12 emplacements en mémoire, pour des mots de 255 caractères au maximum.

Sur le MO6 il n'est pas obligatoire de dimensionner un tableau qui contient moins de 10 éléments.

LA RECHERCHE D'UN ELEMENT DANS UN TABLEAU : R (I)

Si l'on a un tableau à une dimension, numérique ou alphanumérique, défini par une instruction comme DIM R (3Ø) ou DIM N$ (2Ø), le Ième élément de ce tableau est repéré par R (I) (ou N$ (I)) ; il sera alors manipulé exactement comme une variable ordinaire (affectation, calcul, tests, etc.).

Ainsi INPUT R (3) permet d'entrer à la main la valeur du 3e élément numérique du tableau R. PRINT R$ (5) permet l'affichage du 5e élément du tableau alphanumérique R$, etc.

LA RECHERCHE D'UN ELEMENT DANS UNE CHAINE : MID$

Pour extraire le Ième élément de la chaîne alphanumérique A$, on utilisera l'instruction MID$.

Ainsi, si A$ = "BONJOUR", MID$ (A$, 3, 1) donne le caractère N ; de plus MID$ (A$, 3, 3) donne la sous-chaîne "NJO".

LE TRAITEMENT DE DONNEES MULTIPLES INSTRUCTIONS READ, DATA, RESTORE

Pour remplir facilement un tableau, numérique ou alphanumérique, il sera intéressant d'utiliser des lignes DATA, qui contiennent les données inscrites les unes à la suite des autres, séparées par une virgule.

Ainsi, on pourra écrire :

DATA LEON, JULES, 13, rue MARTIN...

Ces données peuvent être lues dans des boucles FOR... NEXT, au moyen de l'instruction READ. On peut recommencer la lecture au début des DATA, à l'aide de l'instruction RESTORE.

Dé truqué

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

LE JEU :
On veut essayer de fabriquer un dé truqué selon sa propre convenance. La méthode proposée ici consiste à définir au départ une probabilité de réalisation pour chaque type de tirage (soit la probabilité d'obtenir un 1, un 2, un 3, etc. que chacune de ces probabilités soit comprise entre Ø et 1, et que la somme de toutes les probabilités soit un nombre très voisin de 1).

LE PROGRAMME :
Il utilise le générateur aléatoire du MO6 supposé de qualité convenable (on pourra aisément utiliser tout autre générateur, si besoin est). La boucle I (7Ø-14Ø) s'occupe de l'entrée de chacune des probabilités, avec vérification à chaque entrée et sur le total (ligne 11Ø) ; chaque probabilité est conservée dans une variable indicée E (I) : ligne 9Ø du programme.
Le lancer de dé truqué est effectué à l'aide d'une autre boucle I (18Ø-2ØØ), la ligne permettant le truquage se trouvant en 19Ø. Un nouveau tirage pourra être effectué sur pression de ENTREE (ligne 22Ø).

POUR JOUER :
Après le lancement de l'exécution par RUN, on obtient l'affichage de la question PROBABILITE DU DE 1, à laquelle on devra répondre par un nombre compris entre Ø et 1. Il conviendra de ne pas oublier que le total de toutes les probabilités entrées ne devra en aucun cas dépasser 1.1 (tolérance de Ø.1).
Des messages d'erreurs pourront être affichés et le programme sera éventuellement relancé. Lorsque toutes les probabilités auront été définies, le premier tirage sera automatiquement effectué, et il suffira de presser la touche ENTREE pour effectuer les tirages suivants. On arrête le programme en tapant CNT et C.

EXTENSIONS POSSIBLES :
Il sera évidemment bien tentant d'utiliser sournoisement ce petit programme pour améliorer vos chances de gain dans un jeu de hasard utilisant des lancers de dés. On pourra alors envisager de figer les données définissant le dé truqué dans une variable A$, etc. On pourra aussi envisager de présenter les résultats correspondant à une série de lancers de dés sous une forme spectaculaire, qui mette le truquage en évidence...

```
10 REM DE TRUQUE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 P=0
60 DIM E(6)
70 FOR I=1 TO 6
80 LOCATE 17,3:PRINT"                    "
90 LOCATE 0,3:PRINT"PROBABILITE DU DE ";
I;:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 24
0
120 LOCATE 6*I-6,6:PRINT"DE";I
130 LOCATE 6*I-6,8:PRINT E(I)
140 NEXT I
150 IF P<0.9 THEN BEEP:RUN 40
160 FOR I=1 TO 500:NEXT I
170 N=INT(RND*6+1)
180 FOR I=1 TO 6
190 IF N=I AND RND>E(I) THEN 170
200 NEXT I
210 LOCATE 0,15:PRINT"TIRAGE :";N;"
      ":PLAY"DO;SO;MI"
220 PRINT:INPUT"FAIRE - ENTREE - POUR UN
 AUTRE TIRAGE";R$
230 GOTO 170
240 LOCATE 0,4:PRINT"IMPOSSIBLE :0<= P <
=1"
250 P=P-E(I)
260 FOR J=1 TO 1500:NEXT J
270 LOCATE 0,4:PRINT"
    "
280 GOTO 90
```

```
PROBABILITE DU DE  6 ? 0.05

DE 1   DE 2   DE 3   DE 4   DE 5   DE 6
 .05    .05    .4     .3     .2     .05



TIRAGE : 3

FAIRE - ENTREE - POUR UN AUTRE
TIRAGE?
```

Visions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

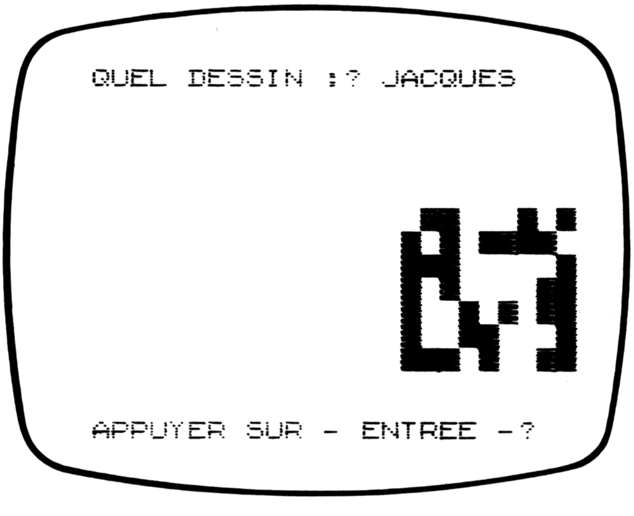
LE JEU :
Le MO6 vous permet, pour la première fois, d'entrer dans son intimité en vous laissant entrevoir sa propre représentation du monde ! Vous allez, en effet, lui proposer des mots, n'importe lesquels, et il va vous répondre en vous proposant une représentation graphique de ce que représente ce mot pour lui. Le jeu sera beaucoup plus amusant si vous le pratiquez à plusieurs, et vous pourrez quelquefois avoir des surprises.

LE PROGRAMME :
Une boucle I (4Ø-12Ø) se charge de réaliser les dessins.
La ligne 5Ø permet de transformer en nombre chaque lettre du mot entré au début (ligne 3Ø). Ce nombre sera ensuite traduit en base 2 par une boucle J (lignes 6Ø à 11Ø) ; selon le résultat de cette transformation, un point sera imprimé, ou non (ligne 9Ø).

POUR JOUER :
On débute l'exécution du programme par RUN, ce qui amène l'affichage de la question QUEL DESSIN. On répond en entrant un mot quelconque qui sera transformé en dessin par le MO6 dès que l'on aura frappé ENTREE.
Pour recommencer, il suffira de frapper une touche quelconque, et CNT et C pour finir. Je vous laisse savourer le dessin qu'il a réalisé lorsqu'il a connu le prénom de l'auteur de ce livre.

EXTENSIONS POSSIBLES :
La même idée peut être reprise en introduisant des éléments de symétrie dans les dessins, en ajoutant la couleur. On peut aussi imaginer de coder par cette méthode un certain nombre de dessins, que l'on introduirait au fur et à mesure dans la machine et l'on pourra alors créer un jeu dans lequel MO6 deviendra de plus en plus savant. On peut aussi réaliser un jeu à deux joueurs dans lequel MO6 vous demande de trouver le mot qui correspond à un certain dessin, etc.

```
10 REM VISIONS
20 CLS
30 INPUT"QUEL DESSIN :";R$
40 FOR I=1 TO LEN(R$)
50 N=9*ASC(MID$(R$,I,1))-340
60 J=1
70 S=N/2-INT(N/2)
80 N=INT(N/2)
90 IF S<>0 THEN LOCATE J+15,I+5:COLOR 3,
1:PRINT" ":COLOR 0,6
100 J=J+1
110 IF J<10 THEN 70
120 NEXT I
130 PRINT:PRINT
140 INPUT"APPUYER SUR - ENTREE -";R$
150 GOTO 20
```



421

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : moyenne.

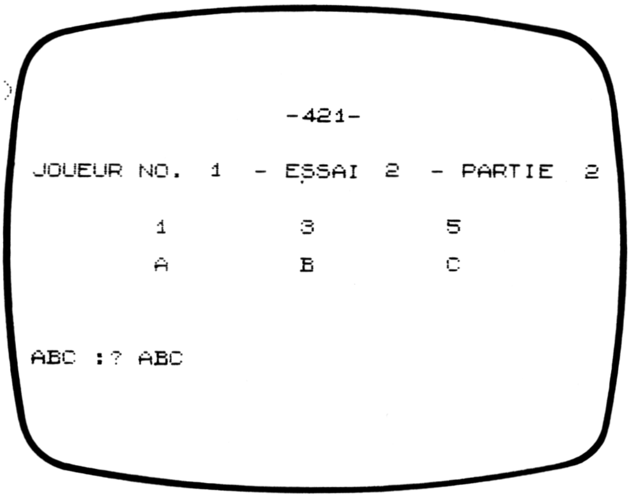
LE JEU :
Chacun des joueurs devra, à son tour, lancer 3 dés, et peut ensuite choisir, sur les deux coups suivants, s'il désire relancer ou non 1, 2 ou 3 dés. Il marque alors un score correspondant au nombre indiqué par les dés, suivant les règles choisies, et le joueur suivant procède de la même manière.

LE PROGRAMME :
Le nombre de joueurs est tout d'abord entré en ligne 4Ø.
La ligne 13Ø affiche le numéro de joueur et le nombre d'essais déjà effectués.
Un sous-programme placé en ligne 4ØØØ, utilisé 3 fois dans la boucle J (2ØØ-25Ø), réalise le lancer de dés.
Les retirages sont effectués par les lignes 1ØØØ, 2ØØØ et 3ØØØ.
La variable R$ gère les 2ᵉ et 3ᵉ tirages. L'impression du résultat est réalisée dans tous les cas par les lignes 14Ø-15Ø.
Les tests pour le nombre d'essais et pour le changement de joueur sont effectués aux lignes 26Ø et 28Ø.

POUR JOUER :
On lance l'exécution du programme par RUN ce qui conduit à l'affichage NOMBRE DE JOUEURS. On entre le nombre de joueurs, et l'on tape ENTREE. L'écran s'efface et affiche JOUEUR N° 1. ESSAI A, PARTIE P, ainsi que le tirage réalisé, et les noms des dés (A, B, C).
Si l'on ne désire pas refaire de tirage, il suffit de frapper ENTREE. La main passe alors au joueur suivant.
Pour effectuer un autre tirage, il suffit de taper le nom du ou des dés à retirer, et de faire ENTREE : par exemple, pour retirer A et C on tapera AC suivi de ENTREE.
Les joueurs suivants procéderont de la même façon.

EXTENSIONS POSSIBLES :
La plus évidente consiste à assurer la gestion des résultats des joueurs, en se référant aux règles de gain usuelles du jeu du 421. On pourra envisager l'affichage après chaque tour d'un tableau récapitulant les résultats des joueurs.
On pourra aussi améliorer l'affichage des dés (se référer aux divers programmes d'affichage des dés se trouvant dans le même ouvrage).

```
10 REM 421
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 INPUT"NOMBRE DE JOUEURS :";N
50 P=1
60 I=1
70 K=0
80 CLS
90 R$="ABC"
100 GOTO 200
110 K=K+1
120 LOCATE 17,1:COLOR 3,1 :PRINT "-421-"
130 COLOR 0,6:LOCATE 0,4:PRINT"JOUEUR NO
. ";I;" - ESSAI ";K;" - PARTIE ";P
140 LOCATE 7,7:PRINT A :LOCATE 17,7:PRIN
T B:LOCATE 27,7:PRINT C
150 LOCATE 8,9:PRINT"A":LOCATE 18,9:PRIN
T"B":LOCATE 28,9:PRINT"C"
160 R$="":LOCATE 0,14:PRINT"
  "
170 LOCATE 0,14:INPUT"ABC :";R$
180 IF R$="" THEN 270
190 R$=R$+"DDD"
200 FOR J=1 TO 3
210 GOSUB 4000
220 X$=MID$(R$,J,1)
230 X=ASC(X$)-64
240 ON X GOSUB 1000,2000,3000,4000
250 NEXT J
260 IF K<3 THEN 110
270 I=I+1
280 IF I<=N THEN 70
290 P=P+1
300 GOTO 60
1000 A=D
1010 RETURN
2000 B=D
2010 RETURN
3000 C=D
3010 RETURN
4000 D=INT(RND*6+1)
4010 RETURN
```



L'électricien fou

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

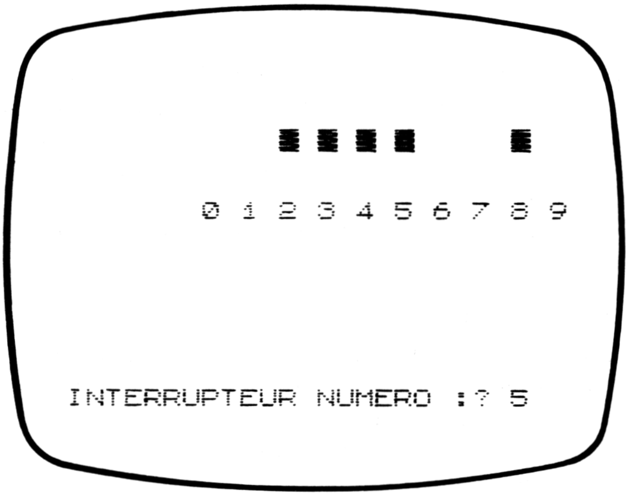
LE JEU :
Un immeuble comportant 1Ø fenêtres, numérotées de Ø à 9, apparaît sur l'écran. Le joueur devra à chaque essai manœuvrer l'un des interrupteurs, en indiquant son numéro (de Ø à 9).
En principe, la fenêtre correspondante devrait s'éteindre, mais il est aussi possible qu'une autre s'allume (ou s'éteigne). Il s'agit, bien sûr, d'essayer d'obtenir l'extinction des 1Ø fenêtres, avec le moins possible d'essais.

LE PROGRAMME :
Une table A, en ligne 5Ø, contiendra l'état (allumé ou éteint) des 1Ø fenêtres.
La boucle I (7Ø-9Ø) et l'autre boucle I (11Ø-16Ø) dessinent l'immeuble dans son état initial.
La ligne 17Ø regarde si tout est éteint, auquel cas l'on ira en ligne 31Ø afficher le nombre d'essais.
La ligne 18Ø attend l'indication de l'interrupteur à manœuvrer.
Les lignes 19Ø à 29Ø définissent le résultat de la manœuvre.

POUR JOUER :
RUN provoque l'affichage de l'immeuble, avec toutes les fenêtres allumées.
A la question INTERRUPTEUR NUMERO, on répond en indiquant le numéro de l'interrupteur à manœuvrer, suivi de ENTREE.
Le résultat de la manœuvre est alors affiché sur l'écran (des fenêtres s'allument ou s'éteignent).
On continue ainsi jusqu'à ce que toutes les fenêtres soient éteintes, et l'on obtient alors l'affichage du nombre d'essais réalisés.

EXTENSIONS POSSIBLES :
On pourra envisager un graphisme très amélioré avec si possible l'utilisation de la couleur. On pourra aussi définir un jeu à deux joueurs, augmenter le nombre de fenêtres, modifier la formule définissant l'allumage ou l'extinction...

```
10 REM L'ELECTRICIEN FOU
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=0 TO F:Y=RND:NEXT I
40 M=0
50 DIM A(10)
60 CLS
70 FOR I=1 TO 7
80 COLOR 3,1:LOCATE 0,I:PRINT"
                                        "
90 NEXT I
100 COLOR 0,6:T=0
110 FOR I=1 TO 10
120 T=T+A(I)
130 LOCATE 2*I+5,3:COLOR 2,4:PRINT" "
140 IF A(I)=0 THEN COLOR 1,7:LOCATE 2*I+
5,3:PRINT" "
150 COLOR 0,6:LOCATE 2*I+4,6:PRINT I-1
160 NEXT I
170 IF T=10 THEN 310
180 LOCATE 0,14:BEEP:INPUT"INTERRUPTEUR
NUMERO :";N
190 N=N+1
200 M=M+1
210 IF A(N)=1 THEN 290
220 A(N)=1
230 R=TAN(RND+N/RND-N)-SIN(RND/N)+336*SI
N(8*N)
240 N=INT(10*(R-INT(R)))
250 IF N=0 THEN N=INT(RND*10+1)
260 IF A(N)=1 THEN 290
270 A(N)=1
280 GOTO 100
290 A(N)=0
300 GOTO 230
310 PRINT:PRINT"TROUVE EN ";M;" ESSAIS"
```



La course au 20

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : logique (deux joueurs, ou plus).
Difficulté du jeu : moyenne.

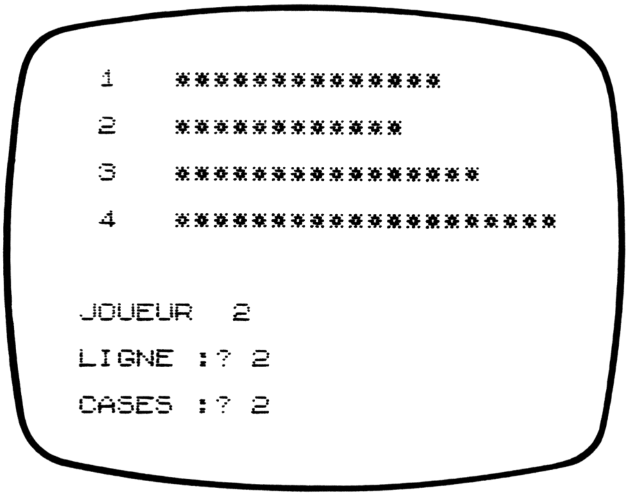
LE JEU :
Il est possible d'imaginer pour ce jeu diverses règles, et nous n'avons retenu ici que la règle usuelle pour deux joueurs, qui en fait un jeu apparenté aux jeux de déplacement, ou d'allumettes : 4 tas de 2Ø étoiles vont apparaître en début de partie, et chaque joueur peut prendre autant d'étoiles qu'il le désire ; mais ceci dans un seul tas à la fois. Celui qui prend la dernière étoile est considéré comme perdant.

LE PROGRAMME :
La boucle I (3Ø-5Ø) dessine 4 rangées de 2Ø étoiles.
La variable F, dimensionnée en ligne 7Ø, servira à enregistrer le nombre d'étoiles prises dans chaque rangée.
La variable T contiendra le nombre total d'étoiles enlevées (ligne 6Ø).
La ligne 12Ø va contrôler qu'il reste suffisamment d'étoiles sur la ligne indiquée.
Les boucles I (15Ø-19Ø) et J (16Ø-18Ø) suppriment les étoiles ; la ligne 22Ø teste la fin de partie.

POUR JOUER :
Lancer le programme par RUN ; le terrain de jeu s'affiche, puis les inscriptions JOUEUR 1 et LIGNES, puis CASES. On répond en entrant un numéro de ligne (entre 1 et 4) suivi de ENTREE, et le nombre d'étoiles à enlever suivi de ENTREE. Les étoiles sont alors supprimées, et c'est au tour du deuxième joueur.
La partie se termine, lorsque la dernière étoile a été enlevée, par l'affichage du joueur gagnant.

EXTENSIONS POSSIBLES :
Outre une amélioration de la qualité du graphisme et des contrôles, de nombreuses variantes sont envisageables, en modifiant si besoin est, les règles de départ : on peut autoriser l'enlèvement d'un nombre impair d'étoiles seulement, ou alterner pair et impair, ou tirer au hasard le nombre à enlever, ou encore jouer à plusieurs...

```
10 REM LA COURSE AU 20
20 CLS
30 FOR I=1 TO 4
40 LOCATE 0,2*I:PRINT I;"  *************
*******"
50 NEXT I
60 T=0
70 DIM F(4)
80 B=1
90 LOCATE 0,12:PRINT"JOUEUR ";B
100 PRINT:INPUT"LIGNE :";L
110 PRINT:INPUT"CASES :";C
120 IF C+F(L)>20 THEN 90
130 F(L)=F(L)+C
140 T=T+C
150 FOR I=1 TO 4
160 FOR J=F(I)-C TO F(I)
170 LOCATE 25-J,2*I:PRINT"  ":BEEP
180 NEXT J
190 NEXT I
200 B=B+1
210 IF B=3 THEN B=1
220 IF T<80 THEN 90
230 PRINT"LE JOUEUR ";B;" GAGNE":PLAY"DO
;SO;MI"
```



Musique

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu éducatif.
Difficulté du jeu : moyenne.

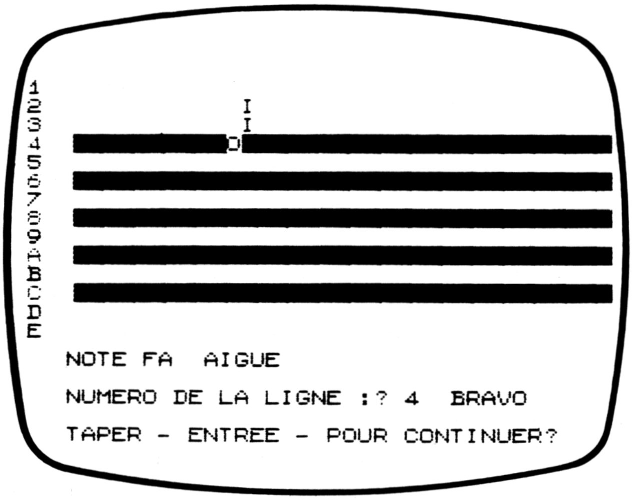
LE JEU :
Il se présente comme un exercice d'apprentissage du solfège : l'appareil annonce une note (en clé de sol), suivie de la mention GRAVE ou AIGUE, selon sa place sur la portée. Il faudra répondre en indiquant sur quelle ligne la note sera affichée.
La note est alors dessinée sur la portée, suivie d'un message BRAVO ou PERDU, selon que l'on avait trouvé juste ou non.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. Une deuxième boucle I (8Ø-12Ø) affiche les numéros des lignes.
La ligne 13Ø définit le nom des notes, et la ligne 14Ø réalise le tirage aléatoire d'une note.
Le nom de la note est calculé à la ligne 15Ø et il est affiché par la ligne 16Ø, suivi de la mention GRAVE ou AIGUE, selon le cas.
La réponse sera entrée en ligne 19Ø.
La ligne 2ØØ s'occupe du décodage de la réponse.
La ligne 21Ø affiche la note.
Les lignes 24Ø et 25Ø regardent si l'on a gagné ou perdu et affichent le message correspondant.
La ligne 26Ø permet de faire un nouvel essai.

POUR JOUER :
On lance le programme par RUN.
On obtient alors l'affichage de la portée, ainsi que le nom de la note à placer et la question NUMERO DE LA LIGNE.
On répond en donnant un numéro de ligne (de 1 à E) suivi de ENTREE.
La note est alors dessinée et le message BRAVO ou PERDU apparaît, dans ce cas, à l'écran.
On doit taper la touche ENTREE pour continuer, et s'arrêter en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut étendre à une autre clé que la clé de sol, on peut introduire de la couleur, des sons, etc.

```
10 REM MUSIQUE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 FOR I=1 TO 5
60 LOCATE 4,2*I+4:COLOR 1,6:FOR J=1 TO 3
5:PRINT CHR$(127);:NEXT J:COLOR 4,6
70 NEXT I
80 FOR I=1 TO 14
90 LOCATE 0,I+2
100 IF I<=9 THEN PRINT I
110 IF I>9 THEN PRINT" ";CHR$(I+55)
120 NEXT I
130 N$="SI LA SOLFA MI RE DO "
140 X=INT(RND*14+1)
150 T=X-1-7*INT((X-1)/7)
160 LOCATE 0,18:PRINT"NOTE ";MID$(N$,3*T
+1,3);
170 IF X<=7 THEN PRINT" AIGUE"
180 IF X>7 THEN PRINT " GRAVE"
190 PRINT:INPUT"NUMERO DE LA LIGNE :";L$
200 L=-(ASC(L$)-48)*(L$<":")-(ASC(L$)-55
)*(L$>="A")
210 LOCATE 15,X:PRINT"I":LOCATE 15,X+1:P
RINT"I":LOCATE 14,X+2:PRINT"O"
230 LOCATE 25,20
240 IF L=X THEN PRINT"BRAVO":IF X>7 THEN
 PLAY "O3"+MID$(N$,3*T+1,2) ELSE PLAY "O
4"+MID$(N$,3*T+1,2)
250 IF L<>X THEN PRINT"PERDU"
260 PRINT:INPUT"TAPER - ENTREE - POUR CO
NTINUER";R$
270 GOTO 40
```



Poker

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu contre le MO6.
Difficulté du jeu : facile.

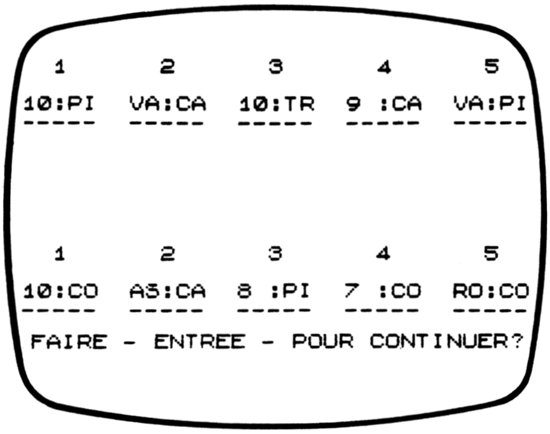
LE JEU :
C'est une simulation d'une partie de poker traditionnelle, effectuée avec un jeu de 32 cartes, dans laquelle le joueur aura, pour partenaire, le MO6. Le joueur recevra 5 cartes, repérées par des numéros de 1 à 5, et pourra décider de changer certaines cartes en annonçant celles qu'il écarte. Sa nouvelle donne sera alors affichée, et c'est ensuite au tour du MO6 de jouer.

LE PROGRAMME :
Les lignes 5Ø et 6Ø définissent les cartes. Une variable N est dimensionnée à 16, en ligne 4Ø, pour recevoir 1Ø cartes joueur et 5 cartes MO6.
Le tirage sera effectué par une boucle I (8Ø-14Ø) et le contrôle par une boucle J (11Ø-13Ø).
L'affichage des cartes est réalisé par le sous-programme 5ØØ-61Ø.
Une autre boucle I (19Ø-21Ø) réalise le deuxième tirage du joueur, la ligne 24Ø permettant le deuxième affichage, et la boucle I (5ØØ-57Ø) où l'on entre directement après, réalisera l'affichage du jeu machine.

POUR JOUER :
L'ordre RUN lance l'exécution du programme, et les 5 premières cartes du joueur sont alors affichées.
A la question CARTES REFUSEES, le joueur peut répondre Ø suivi de ENTREE s'il veut arrêter la partie. Pour jouer, il répondra simplement ENTREE s'il s'estime servi, ou il entrera les numéros des cartes qu'il refuse suivis de ENTREE.
Par exemple, s'il tape 235 et ENTREE, son nouveau jeu conservera les cartes 1 et 4. Son jeu sera alors affiché ainsi que celui du MO6, ce dernier n'ayant pas, dans ce programme, la possibilité de reprendre des cartes.
Il suffira ensuite de faire ENTREE pour démarrer une autre partie.

EXTENSIONS POSSIBLES :
Elles seront, bien sûr, très nombreuses, depuis un affichage graphique en couleur des cartes jusqu'à l'amélioration du jeu du MO6, l'introduction des règles sur les paris et même, pourquoi pas, des possibilités de bluff.

```
10 REM POKER
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 DIM N(16)
50 V$="7 8 9 10VADAROAS"
60 C$="TRCACOPI"
70 CLS
80 FOR I=1 TO 16
90 N(I)=INT(RND*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF N(I)=N(J) THEN 90
130 NEXT J
140 NEXT I
150 K=0:P=0
160 LOCATE 0,12:PRINT"CARTES REFUSEES"
170 GOSUB 500
180 IF R$="0" THEN END
190 FOR I=1 TO LEN(R$)
200 N(VAL(MID$(R$,I,1)))=N(I+5)
210 NEXT I
220 CLS
230 P=1
240 GOSUB 500
250 K=K+10
500 FOR I=1 TO 5
510 J=2*INT(N(I+K)/8)+1
520 F=2*(N(I+K)-8*INT(N(I+K)/8))+1
530 R$=MID$(V$,F,2)
540 LOCATE 7*I-4,K:COLOR 1-2*(K=10): PRI
NT I:BEEP:COLOR 4
550 LOCATE 7*I-5,K+2:PRINT R$;":";MID$(C
$,J,2)
560 LOCATE 7*I-5,K+3:PRINT "-----"
570 NEXT I
580 IF P=1 THEN P=0:RETURN
590 IF K=10 THEN PRINT:INPUT"FAIRE - ENT
REE - POUR CONTINUER";R$:GOTO 70
600 LOCATE 18,12:INPUT R$
610 RETURN
```



Le jeu de saute-mouton

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : facile.

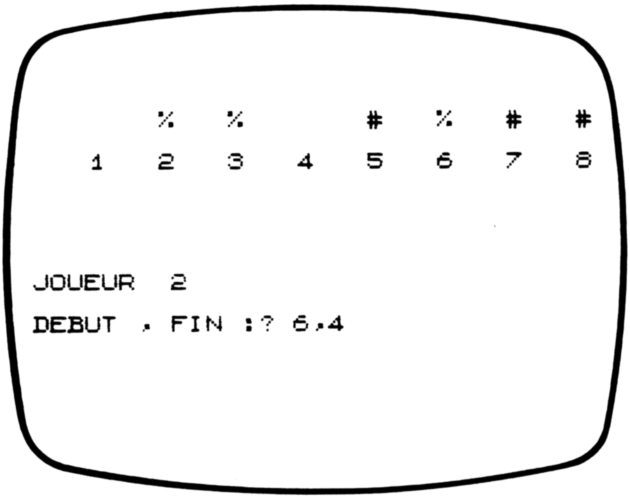
LE JEU :
Bien que l'idée en soit très simple, il ne sera pas forcément évident de découvrir une stratégie convenable. On dispose de 3 moutons noirs et de 3 moutons gris, représentés par des symboles graphiques, qui occupent 8 cases ; les moutons peuvent se déplacer d'une case à une case libre. Plusieurs règles pourront intervenir, selon votre propre estimation de l'intérêt de telle ou telle : soit l'on contraint les moutons à se placer sur la première case disponible, en sautant éventuellement par-dessus les moutons adverses, soit on leur laisse choisir leur case d'arrivée, ce qui peut rendre les parties interminables, à vous de choisir.

LE PROGRAMME :
La boucle I (5Ø-8Ø) initialise une table A (I), qui contient les positions des moutons. La boucle I (1ØØ-15Ø) dessine le terrain de jeu et les moutons.
Les lignes 17Ø et 18Ø s'occupent du texte et des entrées.
La ligne 19Ø contrôle la validité du coup. Le changement de joueur est réalisé en ligne 22Ø.

POUR JOUER :
On commence l'exécution du programme par RUN. On obtient alors un affichage JOUEUR 1, qui indique le numéro du joueur, suivi de la question DEBUT, FIN.
On répond en indiquant le numéro du pion à déplacer suivi de ENTREE, puis le numéro de la case d'arrivée, suivi de ENTREE.
Le déplacement est alors réalisé ; puis le deuxième joueur procède de la même façon. Le gagnant est en principe celui qui réussit à placer ses trois moutons côte à côte sur les emplacements initialement occupés par l'adversaire.

EXTENSIONS POSSIBLES :
On commencera par rajouter un contrôle du joueur gagnant.
On peut également améliorer la présentation en dessinant de vrais moutons, rendre le jeu plus attractif en les numérotant, et en imposant un certain ordre à l'arrivée.
Enfin, on pourra définir une stratégie pour l'ordinateur, en figeant les règles.

```
10 REM LE JEU DE SAUTE-MOUTON
20 CLS
30 DIM A(8)
40 A(1)=0:A(8)=0
50 FOR I=1 TO 3
60 A(I+1)=1
70 A(I+4)=-1
80 NEXT I
90 H=1
100 FOR I=1 TO 8
110 LOCATE 4*I,8:PRINT I:BEEP
120 LOCATE 4*I+1,6:PRINT " "
130 IF A(I)=1 THEN LOCATE 4*I+1,6:COLOR
0,1:PRINT"#":COLOR 4,6
140 IF A(I)=-1 THEN LOCATE 4*I+1,6:COLOR
 1,2:PRINT "%":COLOR 4,6
150 NEXT I
160 J=1-(H=-1)
170 LOCATE 0,13:PRINT"JOUEUR ";J
180 PRINT:INPUT"DEBUT , FIN :";D,F
190 IF A(D)<>H OR A(F)<>0 THEN 170
200 A(D)=0
210 A(F)=H
220 H=-H
230 GOTO 100
```



Lettres

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

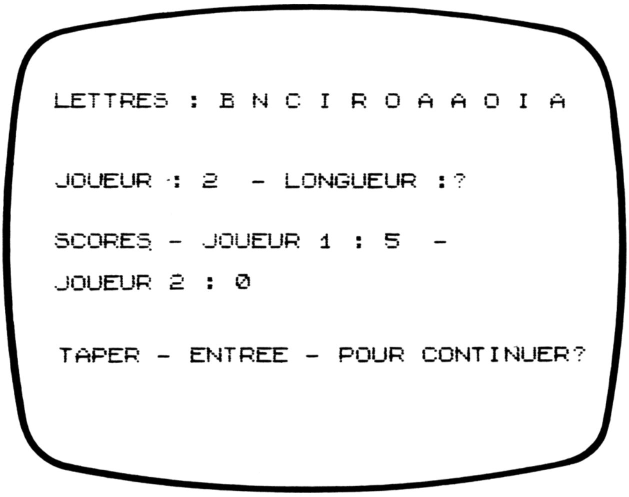
LE JEU :
La règle est fortement inspirée d'un jeu de lettres très connu en France, pratiqué notamment dans une émission télévisée (les chiffres et les lettres).
Le MO6 choisit au hasard un certain nombre de symboles (voyelles et consonnes) dans l'alphabet latin, et chacun des joueurs devra indiquer la longueur du mot qu'il a réussi à obtenir (l'appareil ne vérifiera pas si le mot est correct !).
Le score total de chaque joueur est affiché ensuite.

LE PROGRAMME :
La boucle I (9Ø-12Ø) choisit dans l'alphabet entre 7 et 12 lettres, en fonction de la variable aléatoire L, tirée en ligne 6Ø. Pour éviter d'avoir trop de consonnes, on utilise en ligne 11Ø une petite astuce, qui consiste à refaire au hasard un tirage dans la table A$, contenant des voyelles (on peut en avoir plus ou moins en utilisant d'autres RND).
La boucle 13Ø-18Ø demande à chacun des joueurs la longueur de son mot.
Les lignes 19Ø et 2ØØ comparent les résultats et totalisent le score de chaque joueur, affiché par la ligne 21Ø.

POUR JOUER :
On lance l'exécution du jeu avec RUN.
Les lettres sont alors affichées, puis la question JOUEUR 1 LONGUEUR à laquelle le premier joueur répond en donnant la longueur du mot qu'il a obtenu (l'autre joueur devra toutefois vérifier si le mot est correct) ; puis le joueur 2 fera de même et le score de chaque joueur sera alors affiché.
Pour continuer la partie, il suffira d'appuyer sur la touche ENTREE.
On devra faire CNT et C pour finir.

EXTENSIONS POSSIBLES :
On pourra bien entendu songer à introduire une temporisation avant de demander leurs réponses aux joueurs.
On pourra aussi entrer non point la longueur du mot, mais le mot luimême, l'adversaire devant alors répondre à la question CORRECT OU NON ? On peut aussi étendre le jeu à plus de deux joueurs. Et pourquoi ne pas essayer de définir un jeu contre le MO6, auquel on aurait auparavant entré une liste de mots assez longue, dont il afficherait les lettres dans le désordre, avec éventuellement quelques lettres de complément choisies au hasard ?

```
10 REM LETTRES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 S=0:J=0
50 CLS
60 L=INT(RND*6+7)
70 A$="AEIOUEA"
80 LOCATE 0,6:PRINT"LETTRES :"
90 FOR I=1 TO L
100 LOCATE 8+2*I,6:PRINT CHR$(INT(RND*26
+65)):BEEP
110 IF RND<0.4 THEN LOCATE 8+2*I,6:PRINT
 MID$(A$,INT(RND*7+1),1)
120 NEXT I
130 I=1
140 LOCATE 0,10:PRINT"JOUEUR :";I;:INPUT
" - LONGUEUR :";L
150 LOCATE 25,10:PRINT"     "
160 IF I=1 THEN P=L
170 I=I+1
180 IF I<3 THEN 140
190 S=S+(P AND P>L)
200 J=J+(L AND L>P)
210 LOCATE 0,13:PRINT"SCORES - JOUEUR 1
:";S;" - JOUEUR 2 :";J
220 PRINT:INPUT"TAPER - ENTREE - POUR CO
NTINUER";R$
230 GOTO 50
```



Machinamot

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de hasard (sur des mots).
Difficulté du jeu : aucune.

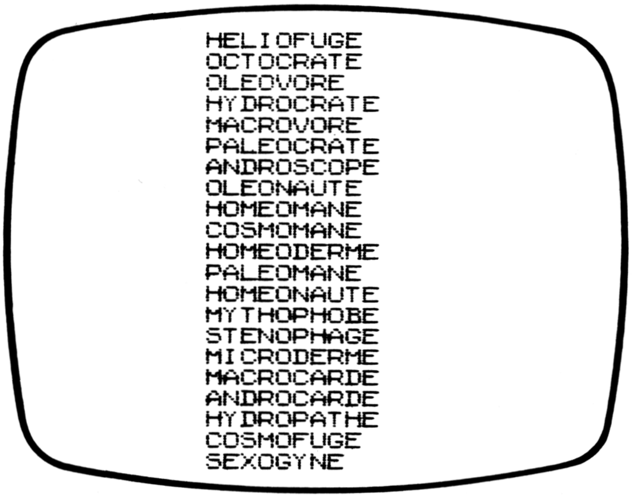
LE JEU :
Ce très court programme permet au MO6 de faire la démonstration de ses capacités à manipuler des chaînes de caractères : on part simplement d'un préfixe et d'un suffixe d'origine grecque, que l'on va assembler au hasard, on tombera bien sûr, de temps à autre, sur des mots absurdes ou même légèrement équivoques.
Un nouveau mot est confectionné à chaque essai, mais il est possible d'obtenir plusieurs fois le même mot.

LE PROGRAMME :
Deux variables A$ et B$ sont dimensionnées à 15 en ligne 5Ø et 6Ø : l'une contiendra les préfixes, l'autre les suffixes (si l'on désire rajouter des possibilités, il suffira de modifier cette limitation à 15).
La boucle I (7Ø-9Ø) lit les lignes DATA (16Ø-18Ø) concernant les préfixes suivis des suffixes.
La ligne 1ØØ effectue le choix d'un préfixe au hasard.
La ligne 11Ø effectue le choix d'un suffixe au hasard.
Le mot obtenu en ligne 12Ø est affiché par la ligne 13Ø, puis on recommence (ligne 15Ø).

POUR JOUER :
Lancer l'exécution du programme par RUN. Un premier mot sera immédiatement affiché, puis un nouveau mot sera affiché à la suite du précédent.
Taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra très facilement augmenter la liste des mots disponibles.
On pourra aussi utiliser un procédé analogue pour faire confectionner des phrases à l'ordinateur, lui faire inventer des poèmes, etc. On pourra aussi transformer le programme initial en un jeu de société qui consisterait, par exemple, à prendre des paris sur le prochain mot qui va apparaître. Et confier, éventuellement, la gestion des paris au MO6...

```
10  REM MACHINAMOT
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 DIM A$(15)
60 DIM B$(15)
70 FOR I=1 TO 15
80 READ A$(I),B$(I)
90 NEXT I
100 I=INT(RND*15+1)
110 J=INT(RND*15+1)
120 C$=A$(I)+B$(J)
130 PRINT C$:BEEP
140 FOR S=1 TO 1200:NEXT S
150 GOTO 100
160 DATA ANDRO,CRATE,COSMO,PHILE,MICRO,P
HOBE,HELIO,TROPE,SEXO,GYNE,STEREO,MANE
170 DATA PALEO,NAUTE,MACRO,PHAGE,HYDRO,F
UGE,STENO,PATHE,PATHO,CARDE,OLEO,DERME
180 DATA MYTHO,SCOPE,HOMEO,VORE,OCTO,PHO
NE
```



Espion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : facile.

LE JEU :
Le programme peut servir simultanément de codeur et de décodeur de message secret, suivant une méthode très facile à mettre en œuvre sur ce type d'appareil.
Un petit générateur aléatoire situé en fin de programme est utilisé pour modifier de façon arbitraire chaque lettre du message ; on devra bien veiller à utiliser toujours le même nombre (noyau) pour commencer le codage ou le décodage de deux messages identiques.

LE PROGRAMME :
Le noyau initial (qui sera un entier quelconque) est entré en ligne 3Ø ; le message est entré à la ligne 5Ø.
La ligne 6Ø permet de choisir entre les options de codage et de décodage.
La boucle I (8Ø-15Ø) est utilisée soit pour effectuer le codage, soit pour effectuer le décodage.
La ligne 1ØØ appelle le générateur aléatoire.
La ligne 11Ø réalise la transposition des lettres.
Les lignes 12Ø et 13Ø vérifient que l'on reste bien dans les limites autorisées.
Les résultats seront imprimés par la ligne 16Ø.
Les lignes 1ØØØ et 1Ø3Ø contiennent le générateur.

POUR JOUER :
Lancer le programme par RUN.
Entrer ensuite un entier N qui servira de noyau pour le générateur (il faudra le retenir pour le décodage).
Puis taper le message, suivi de ENTREE.
Choisir l'option C pour CODER, et l'option D pour DECODER, et taper ENTREE.
Le message codé ou décodé, selon le cas, sera alors affiché.

EXTENSIONS POSSIBLES :
On pourra étendre le codage à d'autres signes que les lettres de l'alphabet, utiliser éventuellement plusieurs générateurs, introduire dans le message un codage du ou des nombres ayant servi de générateurs, etc.

```
10 REM ESPION
20 CLS
30 INPUT"DONNEZ VOTRE CODE :";N
40 A=1/N
50 PRINT:INPUT"VOTRE MESSAGE :";A$
60 PRINT:INPUT"VOULEZ-VOUS (C)ODER OU (D
)ECODER :";R$
70 B$=""
80 FOR I=1 TO LEN(A$)
90 T=ASC(MID$(A$,I,1))
100 GOSUB 1000
110 T=T-A*((R$="C")-(R$="D"))
120 IF T>90 THEN T=T-26
130 IF T<65 THEN T=T+26
140 B$=B$+CHR$(T)
150 NEXT I
160 PRINT:PRINT"RESULTAT :";B$:END
1000 A=(A+3.14159)^8
1010 A=A-INT(A)
1020 A=INT(26*A+1)
1030 RETURN
```

```
DONNEZ VOTRE CODE :? 1357

VOTRE MESSAGE :? BONJOUR MONSIEUR

VOULEZ-VOUS (C)ODER OU (D)ECODER
                              :? C

RESULTAT :JPKKLVO;JPKTFFRS

DONNEZ VOTRE CODE :? 1357

VOTRE MESSAGE :? JPKKLVO JPKTFFRS

VOULEZ-VOUS (C)ODER OU (D)ECODER
:? D
RESULTAT :BONJOUR9MONSIEUR

DONNEZ VOTRE CODE :? 0
```

Accrochez les wagons

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

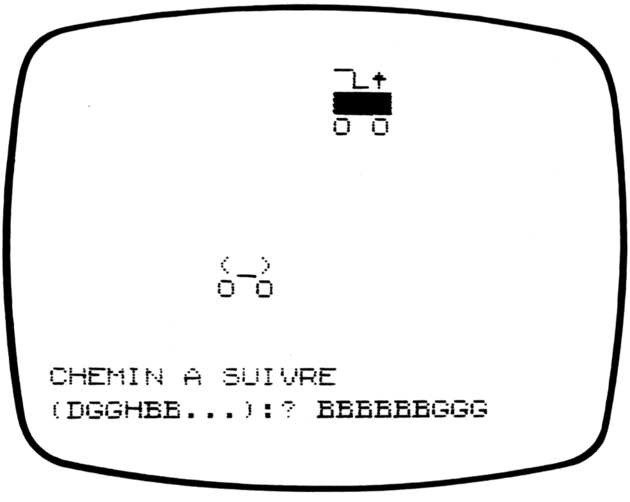
LE JEU :
Il faut essayer d'accrocher, le plus rapidement possible, la locomotive à son wagon, qui apparaissent tous deux en début de partie, en deux endroits plus ou moins éloignés de l'écran. On devra pour cela déplacer la locomotive en indiquant la suite des mouvements à effectuer (sous forme codée : H pour une case vers le haut ; B pour une case vers le bas ; D pour une case vers la droite et G pour une case vers la gauche).
On tapera, par exemple : BBGDHHH.

LE PROGRAMME :
Les lignes 4Ø et 5Ø dessinent la locomotive, la ligne 6Ø le wagon, et les roues sont en ligne 7Ø.
La position initiale de la locomotive est définie par les lignes 9Ø et 1ØØ, celle du wagon par les lignes 11Ø et 12Ø.
La ligne 13Ø évite qu'ils soient initialement trop proches l'un de l'autre.
Les lignes 15Ø et 16Ø réalisent l'affichage du wagon et les lignes 17Ø, 18Ø et 19Ø celui de la locomotive.
Le test de fin est en ligne 2ØØ.
Les mouvements à effectuer sont entrés dans une variable alphanumérique R$, en ligne 21Ø, et sont décodés dans une boucle I (24Ø-27Ø) pour être exécutés en ligne 28Ø.

POUR JOUER :
RUN permet d'afficher les positions de départ de la locomotive et du wagon.
On tape alors la série des déplacements à effectuer, sous la forme DDDHHHBHGG, suivie de ENTREE.
La locomotive sera alors affichée dans sa nouvelle position, et l'on recommence, jusqu'à ce qu'elle soit accrochée au wagon.
On obtiendra alors l'affichage du score.

EXTENSIONS POSSIBLES :
Il sera intéressant de prévoir la perte de la partie si, par accident, au cours du déplacement, la locomotive vient en collision avec le wagon.
On pourra aussi prévoir plusieurs wagons et un ordre d'accrochage, jouer à plusieurs avec plusieurs locomotives, dessiner les voies ferrées, introduire des obstacles à contourner obligatoirement d'un certain côté, etc.

```
10 REM ACCROCHEZ LES WAGONS
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 D$=CHR$(126)+CHR$(76)+"^"
50 A$=CHR$(127)+CHR$(127)+CHR$(127)
60 B$="<"+CHR$(95)+">"
70 C$="O O"
80 N=0
90 LY=INT(RND*33)
100 LX=INT(RND*20+1)
110 WY=INT(RND*33)
120 WX=INT(RND*20+1)
130 IF ABS(LX-WX)<4 THEN 90
140 CLS
150 LOCATE WY,WX:COLOR 1,6:PRINT B$
160 LOCATE WY,WX+1:COLOR 0,6:PRINT C$
170 LOCATE LY,LX-1:COLOR 3,6:PRINT D$
180 LOCATE LY,LX:COLOR 3,6:PRINT A$
190 LOCATE LY,LX+1:COLOR 0,6:PRINT C$:CO
LOR 4,6
200 IF LX=WX AND WY=LY-3 THEN 300
210 LOCATE 0,22:INPUT"CHEMIN A SUIVRE (D
GGHBB...):";R$
220 X=0:Y=0
230 N=N+1
240 FOR I=1 TO LEN(R$)
250 X=X-(MID$(R$,I,1)="B" AND LX+X<22)+
(MID$(R$,I,1)="H" AND LX+X>0)
260 Y=Y-(MID$(R$,I,1)="D" AND LY+Y<37)+
(MID$(R$,I,1)="G" AND LY+Y>0)
270 NEXT I
280 LX=LX+X:LY=LY+Y
290 GOTO 140
300 LOCATE 0,22:PRINT"BRAVO ; TROUVE EN
";N;" COUPS"
310 PLAY"DO;SO;MI"
```



Tirage d'un dé truqué

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

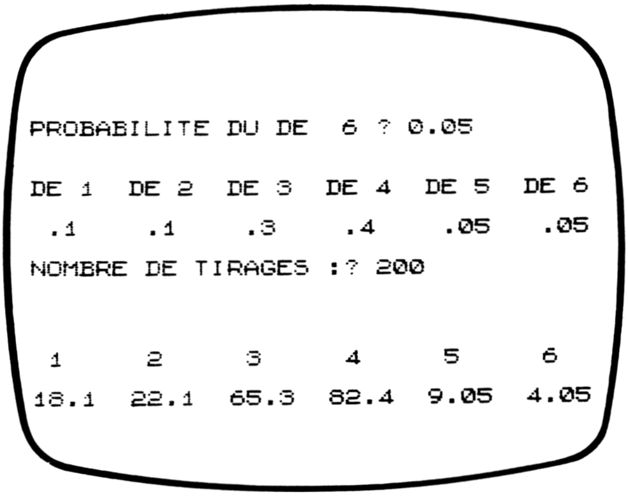
LE JEU :
Il reprend une idée d'un programme précédent (DE TRUQUE) dans une application statistique : on fabrique un dé volontairement truqué, que l'on va lancer un certain nombre de fois, et l'on dresse un tableau des résultats.

LE PROGRAMME :
Les probabilités désirées pour chaque tirage sont introduites par une boucle I (7Ø-14Ø)avec toutes les vérifications de validité nécessaires. Le nombre de tirages demandés est introduit en ligne 16Ø.
La boucle J (17Ø-25Ø) comptabilise les tirages.
La boucle I (19Ø-21Ø) assure la conformité de chaque tirage avec les probabilités introduites en début de jeu.
Les résultats sont comptabilisés en ligne 22Ø, et affichés grâce aux lignes 23Ø et 24Ø.

POUR JOUER :
Commencer l'exécution en frappant RUN, puis à la question PROBABILITE DU DE 1, introduire un nombre compris entre Ø et 1.
Faire de même pour les dés 2, 3, etc., sans oublier que le total des probabilités introduites devra être proche de 1.
Ensuite, à l'affichage de la question NOMBRE DE TIRAGES, on introduira un entier suffisamment grand, suivi de ENTREE. L'affichage des résultats mettra d'autant plus de temps à apparaître que l'entier introduit sera grand.
Les résultats sont affichés sous les numéros des tirages : ainsi 32.3 apparaissant sous le 1 signifie que le tirage de 1, auquel on avait affecté une probabilité de Ø.3, a été réalisé 32 fois.

EXTENSIONS POSSIBLES :
On peut envisager d'améliorer la représentation des résultats en utilisant par exemple un histogramme, ou toute autre méthode graphiquement plus parlante.

```
10 REM DE TRUQUE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 DIM E(6)
50 CLS
60 P=0
70 FOR I=1 TO 6
80 LOCATE 0,3:PRINT"
                  "
90 LOCATE 0,3:PRINT"PROBABILITE DU DE ";
I;:INPUT E(I)
100 P=P+E(I)
110 IF E(I)<0 OR E(I)>1 OR P>1.1 THEN 30
0
120 LOCATE 6*I-6,6:PRINT"DE";:COLOR 1,2:
PRINT I:COLOR 4,6
130 LOCATE 6*I-6,8:PRINT E(I)
140 NEXT I
150 IF P<0.9 THEN RUN
160 PRINT:INPUT"NOMBRE DE TIRAGES :";P
170 FOR J=1 TO P
180 N=INT(RND*6+1)
190 FOR I=1 TO 6
200 IF N=I AND RND>E(I)-INT(E(I)) THEN 1
80
210 NEXT I
220 E(N)=E(N)+1
230 LOCATE 6*N-4,15:COLOR 1,2:PRINT N:CO
LOR 0,6
240 LOCATE 6*N-5,17:PRINT E(N):BEEP
250 NEXT J
260 END
300 LOCATE 0,4:PRINT"IMPOSSIBLE :0<= P <
=1"
310 P=P-E(I)
320 FOR J=1 TO 1700:NEXT J
330 LOCATE 0,4:PRINT"
  "
340 GOTO 90
```



Le jeu de Marienbad

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : logique (2 joueurs).
Difficulté du jeu : moyenne.

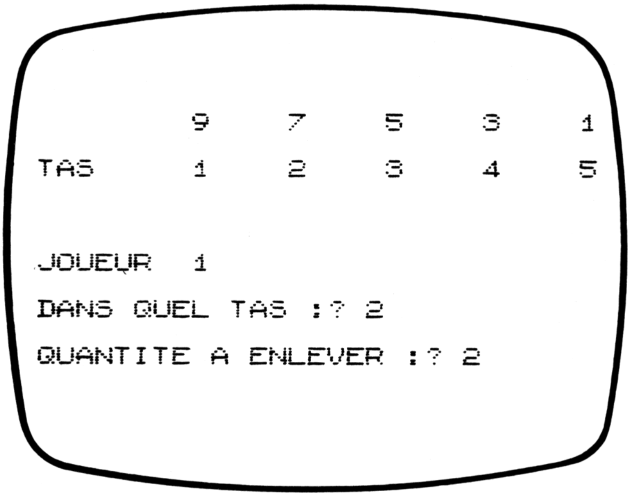
LE JEU :
Un jeu rendu très célèbre grâce à un film qui le popularisa, présenté toutefois ici dans une version plus conforme à ses origines, à savoir le jeu de NIM.
En effet, il est possible, en début de partie, de choisir le nombre d'allumettes que l'on veut mettre dans chaque tas.
Chaque joueur joue à son tour, et celui qui retire la dernière allumette a perdu.

LE PROGRAMME :
La ligne 60 dimensionne une variable A (I) qui contiendra le nombre d'allumettes de chaque tas.
La boucle I (70-100) permet d'entrer ce nombre.
La boucle I (130-160) imprime les tas.
La ligne 170 assure l'affichage du joueur.
La ligne 190, ainsi que la ligne 200, contrôlent si le coup est autorisé.
La ligne 270 contrôle la fin de la partie.

POUR JOUER :
On lance le programme avec RUN, puis on entre le nombre de tas (entre 1 et 8), suivi de ENTREE.
On entrera ensuite le nombre d'allumettes dans chaque tas (faire à chaque entrée N suivi de ENTREE). On obtiendra alors l'affichage des différents tas, avec le numéro du joueur concerné (1 ou 2), ainsi que le message DANS QUEL TAS ; on répond en tapant le numéro du tas dans lequel on désire faire un retrait suivi de ENTREE. On obtient ensuite la question QUANTITE A ENLEVER, et l'on répond en indiquant le nombre d'allumettes que l'on désire enlever du tas concerné.
Puis ce sera au tour du deuxième joueur, et l'on jouera alternativement jusqu'à ce qu'il n'y ait plus d'allumettes disponibles.
En fin de partie, on obtiendra l'affichage du numéro du joueur gagnant.

EXTENSIONS POSSIBLES :
On peut désigner chaque joueur par son nom, faire un graphisme plus élaboré, écrire un programme permettant au MO6 de servir de partenaire, etc.

```
10 REM LE JEU DE MARIENBAD
20 CLS
30 INPUT"NOMBRE DE TAS ( <= 7 ) :";N
40 IF N>=8 THEN RUN
50 S=0
60 DIM A(N)
70 FOR I=1 TO N
80 PRINT:PRINT"TAS ";I;:INPUT" - NOMBRE
D'ALLUMETTES :";A(I)
90 S=S+A(I)
100 NEXT I
110 H=1
120 CLS
130 FOR I=1 TO N
140 LOCATE 5*I+2,4:COLOR 3,1:PRINT A(I)
150 LOCATE 0,6:COLOR 0,6:PRINT"TAS":LOCA
TE 5*I+2,6:PRINT I
160 NEXT I
170 LOCATE 0,10:PRINT"JOUEUR ";H:BEEP
180 PRINT:INPUT"DANS QUEL TAS :";T
190 IF T>N THEN 120
200 IF A(T)=0 THEN 120
210 LOCATE 0,14:INPUT"QUANTITE A ENLEVER
 :";O
220 IF O>A(T) OR O=0 THEN 200
230 A(T)=A(T)-O
240 S=S-O
250 H=H+1
260 IF H=3 THEN H=1
270 IF S=0 THEN PRINT:PRINT"LE JOUEUR ";
H;" A GAGNE":PLAY"DO;SO;MI":END
280 GOTO 120
```



Compose

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : composition musicale.
Difficulté du jeu : aucune, a priori.

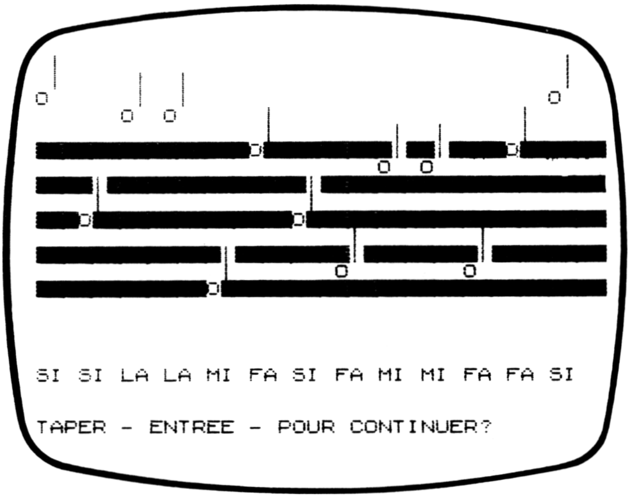
LE JEU :
Il s'agit, en fait, d'un programme capable de composer, de façon très élémentaire, de la musique, et ceci d'une façon complètement aléatoire. L'ordinateur va, dans un premier temps, dessiner une portée, puis il affiche une succession de notes (allant du DO grave au SI aigu) et donne leur nom. Une nouvelle série de notes sera ensuite affichée, et ainsi de suite.

LE PROGRAMME :
La boucle I (5Ø-7Ø) dessine la portée. La variable N$, en ligne 8Ø, contient les noms des notes.
La deuxième boucle I (9Ø-17Ø) permet de faire le choix d'une note de façon aléatoire, à la ligne 1ØØ ; son nom est calculé en ligne 11Ø et la note est dessinée par la ligne 12Ø.
La ligne 18Ø permet de recommencer en tapant ENTREE.

POUR JOUER :
Lancer l'exécution du programme en tapant RUN.
La portée sera alors rapidement affichée, puis les notes sont représentées, avec leur nom affiché en bas de l'écran.
Une pression sur la touche ENTREE provoquera l'affichage d'une nouvelle série de notes, et une pression sur les touches CNT et C provoque la fin de la partie.

EXTENSIONS POSSIBLES :
On pourra affiner les règles de composition, afficher davantage de notes, introduire des dièses et des bémols, améliorer le graphisme, faire jouer la mélodie affichée.

```
10 REM COMPOSE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 FOR I=1 TO 5
60 LOCATE 0,3+2*I:COLOR 1,6:FOR J=1 TO 4
0 :PRINT CHR$(127);:NEXT J
70 NEXT I
80 N$="SI LA SOLFA MI RE DO "
90 FOR I=1 TO 13
100 X=INT(RND*14+1)
110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
120 COLOR 4,6:LOCATE 3*I-2,X-1:PRINT CHR
$(124):LOCATE 3*I-2,X:PRINT CHR$(124):LO
CATE 3*I-3,X+1:PRINT"O"
130 COLOR 0,6
140 LOCATE 3*I-3,18:PRINT D$
150 IF X<=7 THEN PLAY "O5"+LEFT$(D$,2):G
OTO 170
160 PLAY "O4"+LEFT$(D$,2)
170 NEXT I
180 LOCATE 0,21:INPUT"TAPER - ENTREE - P
OUR CONTINUER";R$
190 GOTO 40
```



Jour de la semaine

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : utilitaire.
Difficulté du jeu : facile.

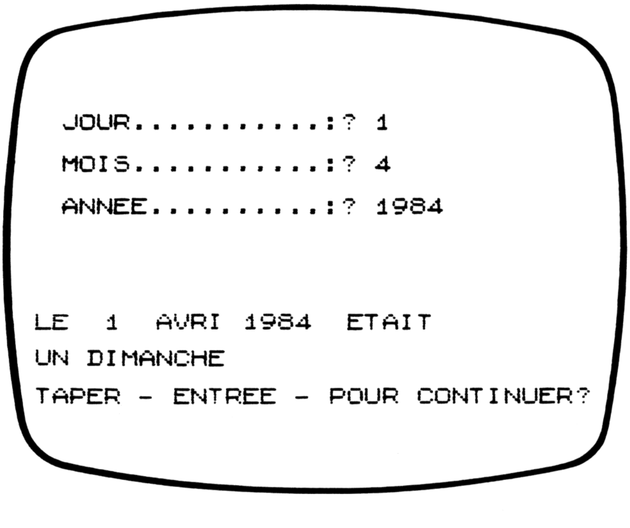
LE JEU :
Le programme vous permet de retrouver, sans aucun calcul, le jour de la semaine correspondant à une date donnée.
La date sera entrée sous forme de trois nombres, par exemple 11, 3, 1956 pour le 11 mars 1956, et le résultat sera affiché en clair.

LE PROGRAMME :
Les lignes 1Ø à 7Ø contrôlent les entrées. La ligne 4Ø s'occupe de la fin de partie. La ligne 9Ø permet une correction pour les dates antérieures à 1582 (mise en service du calendrier actuel).
Les lignes 1ØØ à 11Ø permettront l'affichage en clair du mois dans la réponse (sur 4 lettres).
La formule utilisée apparaît aux lignes 13Ø et 14Ø.
Les lignes 15Ø et 16Ø permettront l'affichage en clair du jour dans la réponse. La ligne 17Ø imprime les résultats.
Le sous-programme 2ØØ-22Ø intervient pour tenir compte des mois de janvier et février.

POUR JOUER :
Lancer le programme par RUN, puis entrer le jour suivi de ENTREE, le mois (entre 1 et 12) suivi de ENTREE, et l'année (sous la forme 1845, par exemple) suivie de ENTREE.
Le résultat est très rapidement affiché.
On tape ENTREE et le programme revient sur la question JOUR.
Pour continuer, il suffit de retaper un jour, un mois, une année.
Pour terminer, il faut répondre Ø à la question JOUR.

EXTENSIONS POSSIBLES :
On pourra entrer les mois en clair et faire afficher le nom du jour en entier.
On peut envisager de faire afficher les dates et fêtes particulières (Noël, Pâques, les Rameaux, Pentecôte...).
On peut écrire un programme qui afficherait les jours fériés pour une année donnée, etc.

```
10 REM JOUR DE LA SEMAINE
20 CLS
30 INPUT"JOUR...........:";J
40 IF J=0 THEN END
50 PRINT:INPUT"MOIS...........:";M
60 PRINT:INPUT"ANNEE..........:";A
70 B=A
80 S=INT(A/100)-INT(A/400)
90 IF A<=1582 THEN S=2
100 M$="JANVFEVRMARSAVRIMAI JUINJUILAOUT
SEPTOCTONOVEDECE"
110 M$=MID$(M$,4*M-3,4)
120 IF M<3 THEN GOSUB 200
130 S=J-1+INT(13*(M+1)/5)+INT(5*A/4)-S
140 S=S-7*INT(S/7)
150 J$="DIMANCHELUNDI   MARDI   MERCREDI
JEUDI   VENDREDISAMEDI  "
160 J$=MID$(J$,8*S+1,8)
170 LOCATE 0,15:COLOR 3,1:PRINT"LE ";J;"
 ";M$;B;" ETAIT UN ";J$:PLAY"DO;SO;MI"
180 COLOR 4,6:PRINT:PRINT:INPUT"TAPER -
ENTREE - POUR CONTINUER";R$
190 RUN
200 M=M+12
210 A=A-1
220 RETURN
```



Le jeu de la vie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu solitaire (2 versions).
Difficulté du jeu : facile.

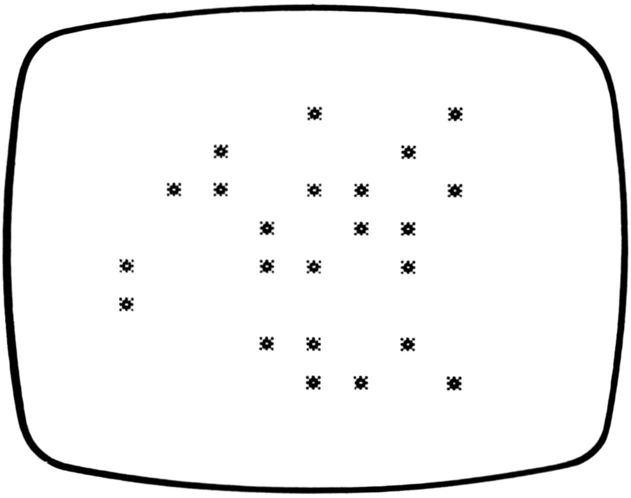
LE JEU :
La version présentée ici respecte les règles posées à l'origine par le mathématicien J.H. CONWAY.
Une certaine population est définie au début du jeu soit par le MO6 (version automatique) soit par vous-même (version manuelle). Cette population va évoluer sous vos yeux, en respectant certaines règles, très élémentaires de naissance, de vie et de mort. L'évolution se poursuivra jusqu'à ce que l'on décide d'arrêter le programme.

LE PROGRAMME :
Deux tables (B et C) sont utilisées en ligne 5Ø, pour enregistrer l'état de la population. Les boucles F (23Ø-31Ø) et G (24Ø-3ØØ) sont utilisées en début de jeu pour définir la population initiale (lignes 25Ø et 26Ø). Le calcul de la nouvelle population est ensuite réalisé à l'aide d'une quadruple boucle : 8Ø-21Ø, 9Ø-2ØØ, 11Ø-16Ø, 12Ø-15Ø. La ligne 13Ø réalise la gestion du nouvel état suivant les règles.
L'affichage de ce nouvel état est obtenu en réutilisant la double boucle d'affichage F (23Ø-31Ø) et G (24Ø-3ØØ).
La ligne 32Ø permet de recommencer.

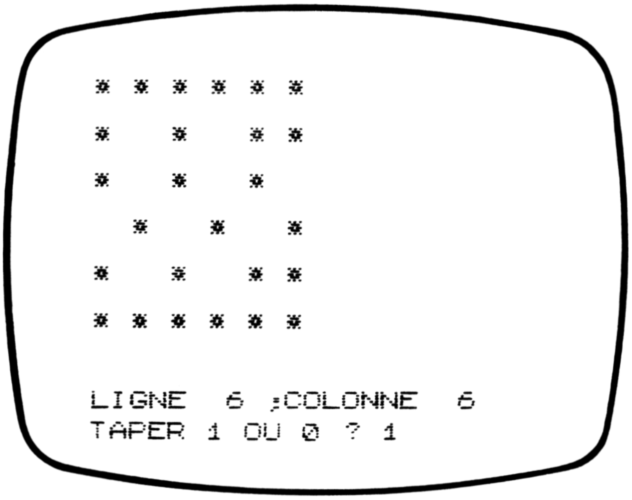
POUR JOUER :
Dans la version automatique, il suffit de lancer le programme par RUN et de regarder. Pour arrêter, il faudra taper CNT et C. Dans la version manuelle, la configuration de départ sera entrée, après le lancement du programme par RUN, en tapant Ø si l'on ne veut rien mettre sur la ligne et la colonne indiquée, et 1 si l'on désire y mettre une cellule. Puis l'évolution se déroulera ensuite de façon automatique.

EXTENSIONS POSSIBLES :
Il sera facile, si l'on dispose d'une mémoire plus importante, d'augmenter les dimensions des tables B et C, en redéfinissant la variable A. On pourra aussi introduire la couleur, améliorer la précision du dessin en remplaçant les PRINT par un affichage haute résolution.

```
10 REM LE JEU DE LA VIE (AUTOMATIQUE)
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 INPUT"DIMENSION :";A
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS:SCREEN 3,1,7
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 270
260 IF RND<0.5 THEN C(F,G)=1
270 LOCATE 2*G,2*F,0:COLOR 3,1:IF C(F,G)
=1 THEN PRINT "*"
280 B(F,G)=C(F,G)
290 NEXT G
300 NEXT F
310 GOTO 80
```



```
10 REM LE JEU DE LA VIE (MANUEL)
40 INPUT"DIMENSION :";A
50 DIM B(A,A):DIM C(A,A)
60 I=0
70 GOTO 220
80 FOR F=1 TO A
90 FOR G=1 TO A
100 S=0
110 FOR I=-1 TO 1
120 FOR J=-1 TO 1
130 IF F+J=0 OR G+I=0 OR F+J=A+1 OR G+I=
A+1 THEN 150
140 S=S+B(F+J,G+I)
150 NEXT J
160 NEXT I
170 S=2*S-B(F,G)
180 C(F,G)=0
190 IF S>=5 AND S<=7 THEN C(F,G)=1
200 NEXT G
210 NEXT F
220 CLS:SCREEN 3,1,7
230 FOR F=1 TO A
240 FOR G=1 TO A
250 IF I<>0 THEN 280
260 LOCATE 0,20:PRINT"LIGNE ";F;";COLONN
E ";G;"TAPER 1 OU 0"
270 LOCATE 0,21:COLOR 0,6:INPUT C(F,G)
280 LOCATE 2*G,2*F:COLOR 3,1:IF C(F,G)=1
 THEN PRINT "*"
290 B(F,G)=C(F,G)
300 NEXT G
310 NEXT F
320 GOTO 80
```



Tour de cartes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de prestidigitation.
Difficulté du jeu : facile.

LE JEU :
C'est un tour assez facile mais, ici, les cartes sont manipulées sous vos yeux par le MO6 qui tient le rôle du prestidigitateur ; 21 cartes, prises au hasard dans un paquet de 52, sont affichées, et le joueur devra retenir mentalement l'une des cartes présentées. Puis, il devra indiquer au MO6 dans quelle colonne (1re, 2e ou 3e à partir de la gauche) se trouve la carte choisie ; les cartes sont alors redistribuées et le joueur indique de nouveau la colonne correspondant à sa carte ; ceci est fait encore une fois et, au 4e affichage, le MO6 est en mesure de deviner la carte choisie.

LE PROGRAMME :
La boucle I (60-120) permet de choisir, au hasard, 21 cartes sur 52, et la boucle emboîtée J (80-100) permet de s'assurer que l'on ne réalise pas deux fois le même tirage. La ligne 140 contient les couleurs des cartes ainsi que les noms des cartes figures (VA, DA, RO, AS), dans une variable alphanumérique B$.
La boucle I (150-240) réalise l'affichage du jeu.
La ligne 160 décode la couleur de la carte.
La ligne 170 choisit la position d'affichage.
La ligne 180 décode la hauteur.
L'affichage proprement dit est effectué de la ligne 190 à la ligne 230.
La ligne 260 permet au joueur d'entrer le numéro de la colonne où se trouve sa carte. La boucle J (300-320) réalise le brouillage, et les boucles emboîtées J (350-390) et I (360-380) fabriquent la nouvelle variable A$ à utiliser pour le prochain affichage.

POUR JOUER :
On lance le programme par RUN.
On obtient un premier affichage de 21 cartes réparties sur 3 colonnes.
On indique alors le numéro (1, 2 ou 3) de la colonne où se trouve la carte choisie, suivi de ENTREE.
On obtient un 2e affichage et l'on fait de même ; puis un 3e affichage et l'on fait de même.
Au 4e affichage, la carte choisie est automatiquement affichée.
On sortira du programme en tapant CNT et C à la place du numéro de la colonne.

EXTENSIONS POSSIBLES :
On peut envisager une amélioration de l'affichage en écrivant les noms des cartes en entier, ou même, en dessinant de véritables cartes, en couleur. On pourra aussi essayer de programmer d'autres tours, utilisant le même principe de base, mais avec davantage de cartes, etc.

```
10 REM TOUR DE CARTES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 A$=""
60 FOR I=1 TO 21
70 B$=CHR$(INT(RND*52+38))
80 FOR J=1 TO LEN(A$)
90 IF B$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A$+B$
120 NEXT I
130 H=3:P=1
140 B$="TRCACOPIVADAROAS"
150 FOR I=1 TO LEN(A$)*P-11*(P=0)
160 S=INT((ASC(MID$(A$,I,1))-38)/13)
170 J=7*INT((I-1)/7)
180 T=ASC(MID$(A$,I,1))-37-13*S
190 IF P=0 THEN LOCATE 0,0:COLOR 0,6:PRI
NT"VOTRE CARTE EST :
   "
200 LOCATE J*P+INT(J/2)-16*(P=0),(I-J)*P
,0:COLOR 1,2
210 IF T<=9 THEN PRINT STR$(T+1);
220 IF T>9 THEN PRINT MID$(B$,ABS(2*T-11
),2);
230 LOCATE (J+3)*P+1+INT(J/2)-21*(P=0),(
I-J)*P:PRINT MID$(B$,2*S+1,2)
240 NEXT I
250 IF P=0 THEN END
260 LOCATE 0,16,0:COLOR 0,1:INPUT"QUELLE
 COLONNE :";N
270 CLS
280 H=H-1
290 R$="20120":B$=""
300 FOR J=0 TO 2
310 B$=B$+MID$(A$,7*VAL(MID$(R$,N+J;1))+
1,7)
320 NEXT J
330 IF H=0 THEN A$=B$:P=0:GOTO 140
340 A$=""
350 FOR J=0 TO 2
360 FOR I=1 TO 7
370 A$=A$+MID$(B$,3*I-2+J,1)
380 NEXT I
390 NEXT J
400 GOTO 140
```

```
 2  CA     4  TR     RO  CA
 9  CA     5  TR      8  TR
VA  PI     8  CO      2  CO
AS  CA    10  CO     VA  TR
 2  PI     4  CO      9  TR
 3  PI    RO  TR      5  CO
RO  CO     5  CA      6  CA




QUELLE COLONNE :? 1
```

L'indésirable

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard et réflexion (1 joueur).
Difficulté du jeu : moyenne.

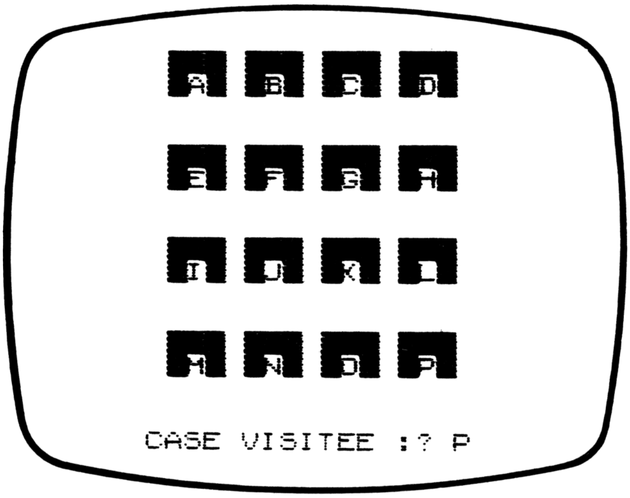
LE JEU :
Vous êtes le propriétaire d'un immeuble de 16 pièces vides, et vous n'arrivez pas à mettre la main sur votre locataire indésirable. Vous visitez les pièces au hasard, mais, si vous visitez la pièce voisine, votre locataire, prévenu par le bruit, va déménager. Il peut même se réinstaller dans une pièce que vous avez déjà visitée, et marquée comme vide. Il vous faudra arriver à le déloger en moins de 9 coups pour gagner.

LE PROGRAMME :
Une table B est dimensionnée à 16, en ligne 60, pour recevoir les numéros des cases déjà visitées.
La boucle I (80-110) dessine le terrain de jeu.
La position initiale du locataire est tirée au hasard en ligne 120.
Le numéro de la case à visiter est donné en ligne 130.
Les lignes 140 à 220 contrôlent les diverses réponses possibles.
La boucle I (180-200) dessine les cases visitées.
La ligne 230 dessine le locataire, effacé en ligne 260.
Les lignes 270 et 280 assurent son déplacement.

POUR JOUER :
On lance l'exécution par RUN.
A la question CASE VISITEE, on répond en tapant une lettre, de A à P.
Si on a trouvé le locataire, celui-ci est affiché, et la partie est terminée ; sinon, deux cas peuvent se produire : soit l'on est suffisamment loin, et le locataire ne bouge pas, soit l'on est à une case, il apparaît un court instant, et il se recache, aléatoirement, dans une case voisine. Les cases déjà visitées se colorent en rouge, et vous ne pouvez pas y revenir. Si, au bout de 9 coups, vous n'avez pas trouvé, la position du locataire s'affiche, et vous avez perdu.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, jouer sur plus de 16 cases, utiliser des caractères graphiques pour représenter le locataire et le propriétaire, utiliser un algorithme plus subtil pour le déplacement du locataire, lui interdire éventuellement de revenir dans des cases déjà visitées, etc.

```
10 REM L'INDESIRABLE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 A$=CHR$(127)
50 B$=A$+A$+A$
60 DIM B(16)
70 CLS
80 FOR I=1 TO 16
90 LOCATE 4+4*(I-1-4*INT((I-1)/4)),2+4*I
NT((I-1)/4):COLOR 7,6:PRINT B$
100 LOCATE 4+4*(I-1-4*INT((I-1)/4)),3+4*
INT((I-1)/4):COLOR 5,6:PRINT A$;:COLOR 0
,6:PRINT CHR$(64+I);:COLOR 5,6:PRINT A$
110 NEXT I
120 X=INT(RND*16+1):S=1:K=0
130 LOCATE 0,20:COLOR 0,6:INPUT"CASE VIS
ITEE :";C$:C=ASC(C$)-64
140 IF B(C)=1 THEN LOCATE 0,22:PRINT"DEJ
A VISITEE":PLAY"DODODO":GOTO 130
150 IF C=X THEN LOCATE 0,22:PRINT"VOUS L
'AVEZ EU EN ";S;" COUPS . BRAVO ":PLAY"D
OMISO":K=1:GOTO 230
160 IF X-5=C OR X-4=C OR X-3=C OR X-1=C
OR X+1=C OR X+3=C OR X+4=C OR X+5=C THEN
 GOTO 230
170 LOCATE 0,22:PRINT"IL N'EST PAS LA":P
LAY"DOREMI"
180 S=S+1:B(C)=1:FOR I=1 TO 16
190 LOCATE 4+4*(I-1-4*INT((I-1)/4)),2+4*
INT((I-1)/4):COLOR 7-6*B(I),6:PRINT B$
200 NEXT I
210 IF S=9 THEN LOCATE 0,22:PRINT"VOUS A
VEZ PERDU . IL ETAIT EN ";CHR$(X+64):K=1
:GOTO 230
220 LOCATE 0,22:PRINT"
                    ": GOTO 130
230 LOCATE 5+4*(X-1-4*INT((X-1)/4)),1+4*
INT((X-1)/4),0:COLOR 1,6:PRINT "%"
240 FOR I=1 TO 300 :NEXT I
250 IF K=1 THEN PLAY"SOMIDO":END
260 LOCATE 5+4*(X-1-4*INT((X-1)/4)),1+4*
INT((X-1)/4),0:COLOR 0,6:PRINT " "
270 Z=INT(10*RND+1)-5:IF ABS(Z)=2 THEN 2
70
280 X=X+Z:IF X>16 OR X<1 THEN X=X-Z:GOTO
 270
290 GOTO 180
```


A B C D
E F G H
I J K L
M N O P
CASE VISITEE :? P

Niveau 3

Nom du programme	*Thèmes et particularités*
Un dé	Tirage d'un dé, avec graphisme.
Plus grand, plus petit	Un jeu de déduction intéressant, pour un joueur.
Le pendu	Un jeu de lettres très classique (2 joueurs).
Belote	Pour simuler une donne, avec un jeu de 32 cartes.
Un taquin de calcul	Un bon entraînement (2 joueurs).
Pendu inversé	Cette fois, c'est l'ordinateur qui essaie de deviner le mot.
Nicomaque	Pour se perfectionner en calcul mental (1 ou plusieurs joueurs).
Course de chevaux	Assez visuel : un jeu de hasard, pour plusieurs joueurs.
A travers	Un jeu de stratégie pour 2 joueurs.
Mastermind	Avec des chiffres, mais très classique. Un seul joueur.
Idem	Un jeu de calcul mental, avec plusieurs niveaux, pour plusieurs joueurs.
Horloge 1	Ne jetez pas votre vieille télévision : transformez-la en horloge.
Reines sur l'échiquier	Un vieux problème, pour vous aider à réfléchir.
Répétitions	Retrouver une séquence de chiffres de plus en plus longue.
Mots	Une idée très simple, mais ce n'est pas si facile de trouver !
Permutations	Un jeu de réflexion, pas toujours évident.
Bridge	Mettez au point vos propres problèmes de bridge.
Hanoï	Un conseil : commencez petit ! ! !
Losanges	Pour vous aider à vous endormir...
Mathieu	Vous ne savez pas dessiner ? Laissez faire votre ordinateur.
Ordre	Un peu d'astuce pour remettre en ordre, le plus vite possible.
Le loup et les agneaux	Un jeu de réflexion pour 2 joueurs.

Manipulation des chaînes et graphiques haute résolution

TRAITEMENT DES CHAINES DE CARACTERES : MID$, LEFT$, RIGHT$

Nous avons eu l'occasion de rencontrer, dès le niveau II, la fonction essentielle utilisée dans le Basic Microsoft pour manipuler les chaînes : il s'agit de MID$ (A$, I J) qui permet d'extraire, dans la chaîne A$, une sous-chaîne de longueur J, commençant au Ième caractère.

Ainsi si,

A$ = "MONSIEUR"

on aura :

MID$ (A$, 2, 3) = "ONS"

Pour faciliter les manipulations, il est possible d'utiliser les deux fonctions LEFT$ et RIGHT$.

LEFT$ (A$, I) permet d'extraire les I premiers caractères à partir de la gauche du mot A$.

Ainsi, LEFT$ (A$, 2) = "MO", avec l'exemple précédent (on pourrait d'ailleurs utiliser MID$ (A$, 1, 2) pour obtenir le même résultat).

RIGHT$ (A$, J) permet d'extraire les J derniers caractères (on part de la droite du mot A$).

Ainsi RIGHT$ (A$, 3) = "EUR", toujours avec le même exemple.

Il est plus difficile de se passer de la fonction RIGHT$, mais il serait possible de faire MID$ (A$, LEN (A$) - 3, 3) pour obtenir le même résultat.

Le Basic du MO6 ne permet pas le remplacement direct d'une sous-chaîne par une affectation du type MID$ (-.-) = " ". Il sera nécessaire, pour remplacer la Ième lettre de la chaîne A$, d'écrire :

A$ = LEFT$ (A$, I – 1) + "nouvelle lettre" + RIGHT$ (A$, LEN (A$) – I).

Des formules analogues pourront être utilisées pour remplacer dans A$ une sous-chaîne par une autre, pas nécessairement de même longueur, ce qui permet une grande souplesse.

AFFICHAGE GRAND FORMAT : ATTRB

Le Basic du MO6 dispose d'une possibilité assez rare qui permettra d'améliorer très facilement l'aspect graphique de certains jeux : on peut, en effet, à l'aide d'une instruction très simple à manipuler, obtenir des caractères expansés, soit dans le sens de la hauteur, soit dans le sens de la largeur, soit même les deux à la fois. Dans ce dernier cas, les caractères sont alors définis dans une matrice 16 × 16, et non plus 8 × 8 ; on ne pourra alors disposer que de 12 lignes de 2Ø caractères.

L'instruction ATTRB s'utilise sous la forme : **ATTRB I, J.**

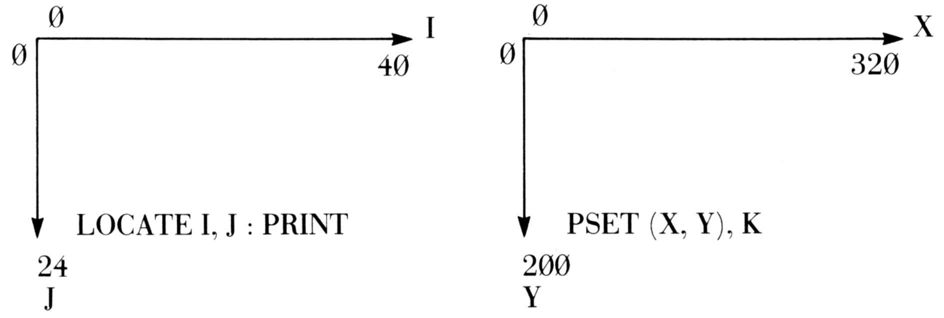
I et J peuvent prendre les valeurs Ø et 1, avec les combinaisons suivantes :

ATTRB Ø, Ø : taille normale.
ATTRB Ø, 1 : largeur normale, hauteur double.
ATTRB 1, Ø : largeur double, hauteur normale.
ATTRB 1, 1 : largeur double, hauteur double.

LES GRAPHIQUES HAUTE RESOLUTION : INSTRUCTION PSET

Le MO6 dispose de la possibilité de mélanger sur l'écran, très facilement, des textes (avec 24 lignes de 4Ø caractères) et des graphiques haute résolution (avec une définition de 2ØØ × 32Ø).

Le système de repérage utilisé est analogue pour le texte, et le graphique :



Le paramètre K indique, s'il est positif, la couleur du point (de Ø à 15), et, s'il est négatif, la couleur du fond (de − 1 à − 16) ; un exemple d'utilisation est donné dans le programme Losanges.

Un dé

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : tirage d'un dé (avec graphisme).
Difficulté du jeu : aucune.

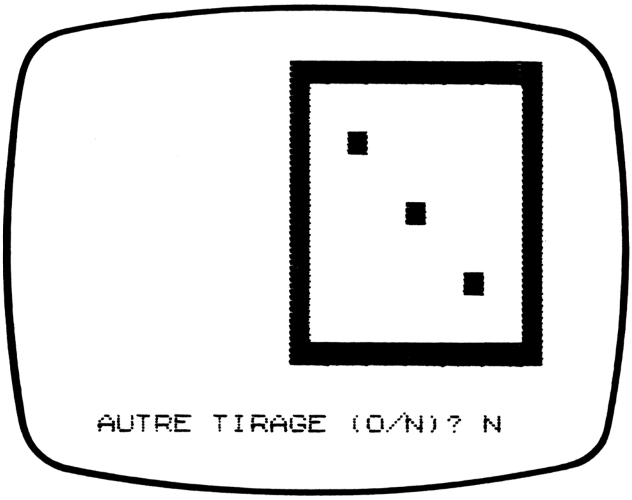
LE JEU :
Le programme propose simplement le lancer d'un dé aléatoire, et sa représentation graphique en grand format.
Il pourra facilement être utilisé en sous-programme, dans tout type de jeux faisant appel au lancer d'un ou de plusieurs dés.

LE PROGRAMME :
La ligne 5Ø utilise le générateur aléatoire du MO6 pour tirer un nombre de 1 à 6.
La boucle J (7Ø-9Ø) dessine les bords du dé, et les lignes 6Ø et 1ØØ dessinent le haut et le bas.
Les lignes 12Ø à 17Ø sont utilisées pour le dessin des points.
Les lignes 18Ø et 19Ø permettront le cas échéant de refaire un tirage.

POUR JOUER :
Lancer le programme par RUN.
On entre un entier pour initialiser le générateur, et un dé s'affiche, en couleurs, et en grand format.
A la question : AUTRE TIRAGE (O/N) on répond O, suivi de ENTREE, pour relancer le dé.

EXTENSIONS POSSIBLES :
On pourra introduire une petite musique agrémentant chaque lancer, faire défiler les points comme si le dé roulait sur une table, etc.

```
10 REM UN DE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN ,4,5
50 A=INT(RND*6+1)
60 COLOR ,1:LOCATE 10,5:PRINT"
   "
70 FOR J=0 TO 10
80 LOCATE 10,6+J:PRINT" ":LOCATE 22,6+J:
PRINT" "
90 NEXT J
100 LOCATE 10,17:PRINT"             "
110 COLOR ,2:PLAY"DO;SO;MI"
120 IF A<>1 THEN LOCATE 13,8:PRINT" "
130 IF A>=4 THEN LOCATE 19,8:PRINT" "
140 IF A=6 THEN LOCATE 13,11:PRINT" ":LO
CATE 19,11:PRINT" "
150 IF A/2 <>INT(A/2) THEN LOCATE 16,11:
PRINT" "
160 IF A>=4 THEN LOCATE 13,14:PRINT" "
170 IF A<>1 THEN LOCATE 19,14:PRINT" "
180 COLOR 4,6:LOCATE 0,20:INPUT "AUTRE T
IRAGE (O/N)";R$
190 IF LEFT$(R$,1)="O" THEN 40
```



Plus grand, plus petit

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de déduction (un seul joueur).
Difficulté du jeu : difficile.

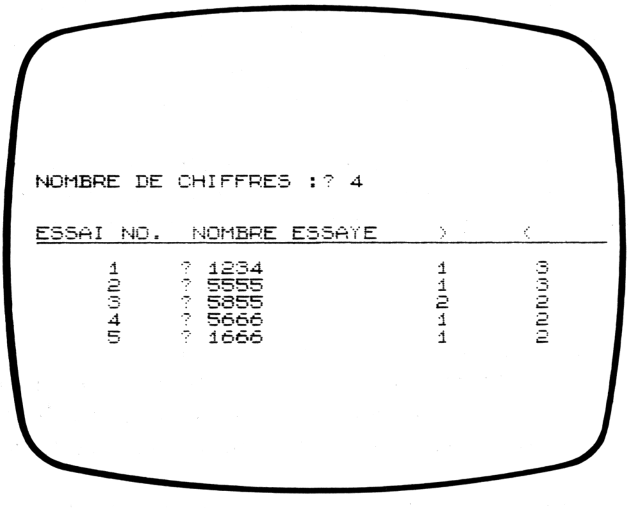
LE JEU :
Il s'inspire d'un principe légèrement différent de celui du Mastermind ; il s'agit également de deviner un nombre (de 1 à 7 chiffres), mais les indications dont on dispose sont cette fois d'une part le nombre de chiffres qui, dans le nombre essayé, sont supérieurs à ceux du nombre à deviner, et, d'autre part, le nombre de chiffres qui sont inférieurs. Si, par exemple, le nombre à deviner est 3Ø54, et que l'on essaie 4251, la machine répondra 2 plus grands (ce sont le 4 et le 2), et 1 plus petit (c'est le 1).

LE PROGRAMME :
La ligne 6Ø interdit de demander des nombres de plus de 8 chiffres.
Le nombre à deviner est placé en ligne 8Ø, dans une variable alphanumérique X$.
Les comparaisons avec le nombre essayé sont faites à la ligne 13Ø (qui teste si l'on a trouvé juste), et dans la boucle I (15Ø-18Ø) où l'on fait la comparaison "lettre à lettre".
L'affichage du résultat est réalisé grâce à la ligne 19Ø.
Le compteur du nombre d'essais est en ligne 2ØØ.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage NOMBRE DE CHIFFRES.
On entre un nombre inférieur à 8 et on appuie sur la touche ENTREE.
Le numéro de l'essai s'affiche, ainsi que la question NOMBRE ESSAYE ; on répond en entrant un nombre, suivi de ENTREE.
Le résultat est alors affiché sous la forme > et <.
S'il n'est pas exact, on recommence (attention à entrer uniquement des nombres ayant autant de chiffres que le nombre à deviner).
Si l'on a trouvé le MO6 affichera VOUS AVEZ TROUVE... EN... ESSAIS.
Puis AUTRE PARTIE (O/N) ; on fera O suivi de ENTREE pour commencer une nouvelle partie.

EXTENSIONS POSSIBLES :
On pourra, comme pour le Mastermind, imaginer un jeu analogue avec des lettres, des couleurs, ou même en utilisant d'autres symbolismes.
On peut aussi prévoir de laisser jouer le MO6 qui tiendra alors le rôle de décodeur.

```
10 REM PLUS GRAND , PLUS PETIT
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 INPUT"NOMBRE DE CHIFFRES :";C
60 IF C>8 THEN 40
70 E=1:F=1
80 X$=STR$(INT(RND*9^C+10^(C-E)))
90 X$=RIGHT$(X$,LEN(X$)-1)
100 LOCATE 0,3:PRINT"ESSAI NO.  NOMBRE E
SSAYE     >      <    "
110 LOCATE 0,4:COLOR,1:FOR J=1 TO 40:PRI
NT CHR$(126);:NEXT J
120 LOCATE 0,F+4:COLOR,6:PRINT TAB(4);E;
TAB(10):INPUT P$
130 IF P$=X$ THEN 220
140 A=0:B=0
150 FOR I=1 TO C
160 IF MID$(P$,I,1)>MID$(X$,I,1) THEN A=
A+1
170 IF MID$(P$,I,1)<MID$(X$,I,1) THEN B=
B+1
180 NEXT I
190 LOCATE 27,F+4:PRINT A;TAB(34);B
200 E=E+1:F=F+1:IF F=18 THEN F=1
210 GOTO 120
220 PRINT:PRINT"VOUS AVEZ TROUVE ";X$;"
EN ";E;" ESSAIS":PLAY"DO;SO;MI"
230 PRINT:INPUT"AUTRE PARTIE (O/N)";R$
240 IF LEFT$(R$,1)="O" THEN 40
```



Le pendu

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de lettres (version deux joueurs).
Difficulté du jeu : moyenne.

LE JEU :
Il serait sans doute assez difficile de retrouver le nom de l'inventeur de ce jeu très à la mode, mais il est certain qu'il existe depuis des temps immémoriaux, quelquefois pratiqué sous des formes assez différentes (par exemple, reconstituer un petit cochonnet en lançant des dés).
L'apparition des jeux sur micro-ordinateur et la magie des écrans de télévision lui ont fait connaître un renouveau certain : dans la version retenue, l'un des joueurs doit entrer un mot de longueur quelconque. L'ordinateur affichera alors des tirets à l'emplacement des lettres de ce mot et demande au deuxième joueur de proposer une lettre. Chaque réponse correcte fait apparaître la lettre à son emplacement exact dans le mot, et chaque réponse fausse fait progresser le dessin de la potence.
Pour gagner, il suffit de réussir à trouver le mot inconnu avant d'être pendu.

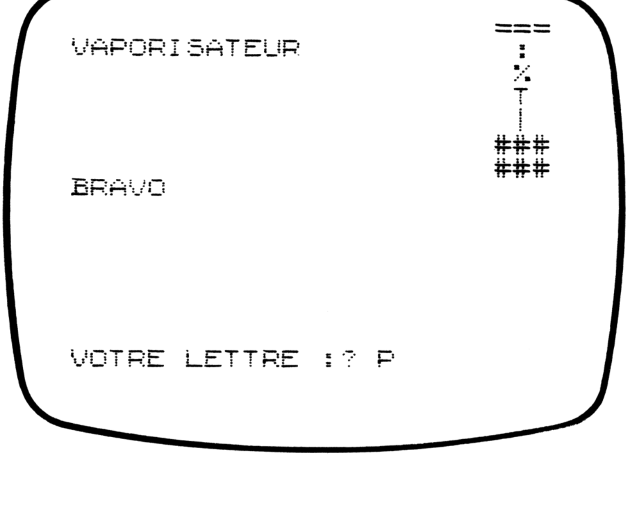
LE PROGRAMME :
Le mot à deviner est introduit en début de programme (ligne 3Ø) ; une variable B$, en ligne 4Ø, contient tous les éléments du dessin.
La boucle I (6Ø-8Ø) fabrique un mot P$ qui contient des tirets à l'emplacement de chacune des lettres du mot initial.
La boucle I (12Ø-14Ø) s'occupe du dessin (plus ou moins complet, selon la valeur de P).
Les lignes 15Ø-16Ø affichent le résultat et arrêtent le jeu.
La lettre essayée est entrée en ligne 17Ø et testée dans une boucle I (19Ø-21Ø) ; le sous-programme 3ØØ-32Ø remplace chaque tiret par la lettre correspondante.

POUR JOUER :
Après le lancement du programme par RUN, l'un des joueurs doit entrer son mot et taper ENTREE. Des tirets sont affichés à l'emplacement de chacune des lettres du mot ; et la question VOTRE LETTRE apparaît sur l'écran ; l'autre joueur répond alors en proposant une lettre suivie de ENTREE.
Si la lettre est bonne, elle viendra remplacer un ou plusieurs tirets ; sinon, la potence va commencer à se dessiner.
Le mot exact sera affiché dans le cas où l'on aurait perdu.

EXTENSIONS POSSIBLES :

Il sera intéressant de constituer une liste de mots dans lesquels le MO6 pourra piocher au hasard ; on pourra aussi améliorer le dessin (en utilisant GR$, par exemple) et modifier le nombre d'essais permis selon la difficulté du mot.

```
10 REM LE PENDU
20 CLS
30 INPUT"VOTRE MOT :";R$
40 B$="< >"+CHR$(124)+" "+CHR$(124)+CHR$
(127)+CHR$(127)+CHR$(127)+"###### "+CHR$
(124)+"  T  %  : ==="
50 P$=""
60 FOR I=1 TO LEN(R$)
70 P$=P$+"-"
80 NEXT I
90 P=11
100 CLS
110 LOCATE 0,3:PRINT P$
120 FOR I=11 TO P STEP -1
130 COLOR 3,1:LOCATE 22,12-P:PRINT MID$(
B$,1+3*(I-1),3):COLOR 4,6
140 NEXT I
150 IF P$=R$ THEN PRINT"BRAVO":PLAY"DO;S
O;MI":END
160 IF P=1 THEN PRINT"PERDU ; LE MOT ETA
IT :";R$:PLAY"DO;DO;DO":END
170 LOCATE 0,20:INPUT"VOTRE LETTRE :";S$
180 T=0
190 FOR I=1 TO LEN(R$)
200 IF MID$(R$,I,1)=S$ THEN GOSUB 300
210 NEXT I
220 IF T<>0 THEN 110
230 P=P-1
240 GOTO 110
300 T=1
310 P$=LEFT$(P$,I-1)+S$+RIGHT$(P$,LEN(P$
)-I)
320 RETURN
```



Belote

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de cartes.
Difficulté du jeu : facile.

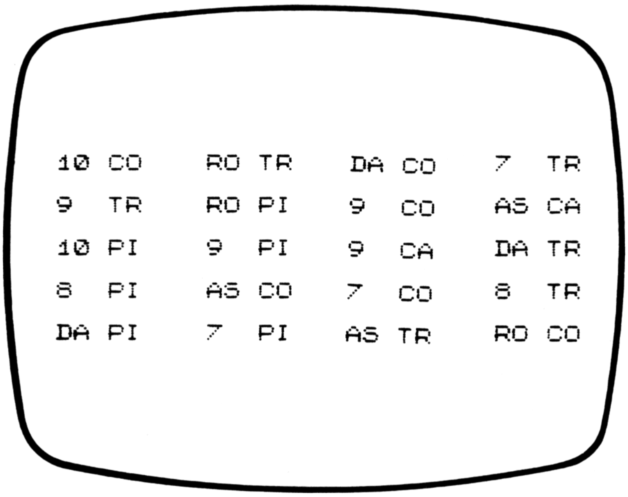
LE JEU :
Le jeu proposé est plutôt une simulation, réalisée sur 32 cartes, de la distribution de 4 paquets de 5 cartes. Il pourra être utilisé pour l'étude de problèmes faisant intervenir des donnes de 2Ø cartes sur 32.

LE PROGRAMME :
En ligne 5Ø, on crée un tableau pouvant contenir les 2Ø cartes.
La boucle I (8Ø-14Ø) assure le tirage aléatoire de 2Ø nombres, pris parmi 32 ; la boucle J (11Ø-13Ø) vérifie que l'on ne réalise pas deux fois le même tirage.
La deuxième boucle I (16Ø-31Ø) assure ensuite l'affichage, les noms des cartes étant affichés en clair, et la couleur étant décodée et affichée en clair, grâce aux lignes 21Ø à 28Ø.

POUR JOUER :
Il suffit de taper RUN pour obtenir immédiatement l'affichage de 4 paquets de 5 cartes chacun (une donne).
Un nouveau RUN permet d'obtenir une nouvelle distribution.

EXTENSIONS POSSIBLES :
On pourra facilement mettre au point un affichage plus graphique en utilisant les caractères graphiques redéfinissables du MO6 ; on pourra également se servir de ce programme comme d'un noyau pour la mise au point de programmes de jeux de cartes sur l'ordinateur (qui pourra, si nécessaire, servir de partenaire : voir l'exemple du Poker, dans le même ouvrage).

```
10 REM BELOTE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 DIM A(20)
60 K=1
70 LOCATE 12,12,0:PRINT"PATIENTEZ , JE T
IRE"
80 FOR I=1 TO 20
90 A(I)=1+INT(RND*32)
100 IF I=1 THEN 140
110 FOR J=1 TO I-1
120 IF A(I)=A(J) THEN 90
130 NEXT J
140 NEXT I
150 CLS
160 FOR I=1 TO 20
170 S$="CA"
180 S=1+A(I)-8*INT(A(I)/8)
190 R$=STR$(S+6)+"  ":R$=RIGHT$(R$,LEN(R
$)-1)
200 R$=LEFT$(R$,2)
210 IF S=5 THEN R$="VA"
220 IF S=6 THEN R$="DA"
230 IF S=7 THEN R$="RO"
240 IF S=8 THEN R$="AS"
250 IF A(I)<=24 THEN S$="TR"
260 IF A(I)<=16 THEN S$="PI"
270 IF A(I)<=8 THEN S$="CO"
280 COLOR 1,2:LOCATE 10*INT((I-1)/5),2*K
:PRINT R$;:COLOR 4,6:PRINT" ";:COLOR 1,2
:PRINTS$:COLOR 4,6
290 K=K+1:BEEP
300 IF K=6 THEN K=1
310 NEXT I
```



Un taquin de calcul

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

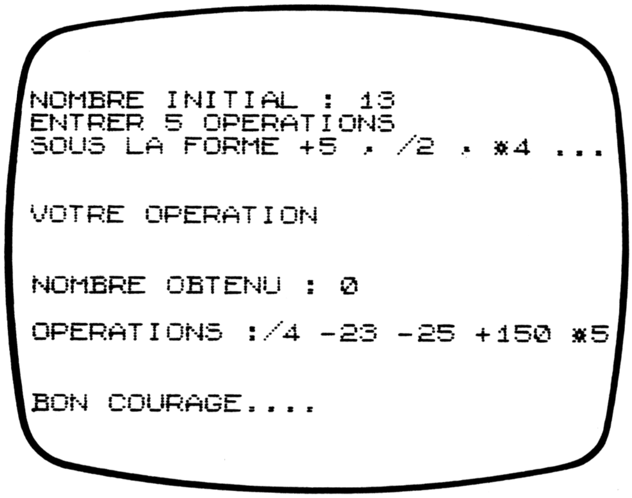
LE JEU :
Il se pratique à deux joueurs, l'un tenant le rôle du codeur, l'autre celui du décodeur.
Le codeur part d'un nombre (proposé par le MO6) et doit le transformer à l'aide de 5 opérations qu'il choisit lui-même, ainsi d'ailleurs que les opérateurs.
Le résultat qu'il a obtenu et les opérations qu'il a utilisées seront affichés sur l'écran : le décodeur devra alors essayer de retrouver le nombre initial en effectuant les opérations en sens inverse.

LE PROGRAMME :
La ligne 5Ø permet d'obtenir l'entier de départ compris entre 5 et 2Ø.
Une boucle I (11Ø-2ØØ) permet d'entrer successivement les 5 opérations.
4 sous-programmes sont utilisés pour les calculs (aux lignes 1ØØØ et 4ØØØ).
Après chaque calcul, le résultat est affiché (seul le codeur devra regarder l'écran pendant cette opération) et le résultat définitif restera affiché pour le décodeur à la fin des calculs (ligne 19Ø).
La ligne 21Ø affiche les 5 opérations utilisées, ainsi que les opérateurs correspondants.

POUR JOUER :
On frappe RUN pour lancer le programme.
Le nombre du début est affiché, puis la question ENTREZ 5 OPERATIONS ; on devra les entrer l'une après l'autre, sous la forme, par exemple : + 4 suivi de ENTREE.
Le résultat sera affiché après chaque opération entrée. A la fin, les opérations entrées sont toutes affichées sur une même ligne, et le message BON COURAGE est envoyé à l'adresse du décodeur (il n'y a pas de contrôle au décodage).

EXTENSIONS POSSIBLES :
On pourra, bien sûr, songer à améliorer la partie décodage du programme en introduisant des contrôles sur le résultat ; on peut aussi envisager de laisser le MO6 participer davantage en lui confiant le rôle du codage, etc.

```
10 REM UN TAQUIN DE CALCUL
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 X=INT(RND*16+5):BEEP
60 PRINT"NOMBRE INITIAL :";X
70 PRINT"ENTRER 5 OPERATIONS"
80 PRINT"SOUS LA FORME +5 , /2 , *4 ..."
90 F=X
100 DIM R$(5)
110 FOR I=1 TO 5
120 LOCATE 0,6:INPUT"VOTRE OPERATION :";
R$(I)
130 LOCATE 16,6:PRINT "           "
140 A=ASC(LEFT$(R$(I),1))-41
150 IF A<=0 OR A=3 OR A=5 OR A>6 THEN 12
0
160 IF A>3 THEN A=(A/2)+1
170 ON A GOSUB 1000,2000,3000,4000
180 LOCATE 14,10:PRINT"         "
190 LOCATE 0,10:PRINT"NOMBRE OBTENU :";F
200 NEXT I
210 PRINT:PRINT"OPERATIONS :";R$(4);" ";
R$(1);" ";R$(5);" ";R$(3);" ";R$(2)
220 PRINT:PRINT:PRINT"BON COURAGE...."
230 PLAY"DO;SO;MI":END
1000 F=F*VAL(RIGHT$(R$(I),LEN(R$(I))-1))
1010 RETURN
2000 F=F+VAL(RIGHT$(R$(I),LEN(R$(I))-1))
2010 RETURN
3000 F=F-VAL(RIGHT$(R$(I),LEN(R$(I))-1))
3010 RETURN
4000 F=F/VAL(RIGHT$(R$(I),LEN(R$(I))-1))
4010 RETURN
```



Pendu inversé

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu sur les mots.
Difficulté du jeu : facile.

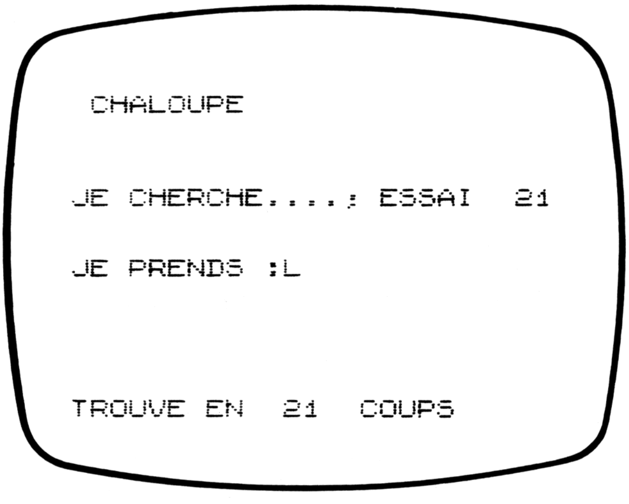
LE JEU :
Le Pendu est un jeu de lettres très connu, présenté par ailleurs dans ce recueil, dans lequel il s'agit de deviner un mot caché par l'ordinateur et où chaque lettre non découverte vous rapproche un peu plus de la potence. Ici, c'est le MO6 qui va chercher à deviner votre mot, mais il joue quelque peu au hasard et ne possède pas votre intuition ; il faudra donc, en règle générale, beaucoup d'essais pour découvrir la solution. Pour augmenter ses chances, on a fait appel à une formule qui lui permet de tirer davantage dans les voyelles puis, dans les consonnes courantes, et, enfin, dans les consonnes plus rares.

LE PROGRAMME :
La ligne 5Ø permet d'enregistrer, dans une variable alphanumérique D$, les lettres de l'alphabet, dans l'ordre : voyelles, consonnes usuelles, autres consonnes. Le mot à deviner est introduit en ligne 7Ø (en fait, il suffirait d'introduire sa longueur).
La boucle I (9Ø-12Ø) imprime des tirets aux emplacements des lettres à trouver.
La boucle en ligne 14Ø sert uniquement à simuler le temps de recherche d'une lettre.
La formule, placée en ligne 15Ø, permet de modifier légèrement la probabilité de tirage, en faveur des voyelles et des consonnes usuelles.
La ligne 17Ø provoque un nouveau tirage, et la boucle I (21Ø-24Ø) réimprime les tirets et les lettres trouvées.
La ligne 25Ø incrémente le compteur d'essais.
La ligne 26Ø affiche le nombre d'essais réalisés, en fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Entrer le mot à découvrir, suivi de ENTREE.
Des tirets sont alors affichés (un par lettre) et le MO6 propose une première lettre, regarde si elle convient et continue automatiquement, jusqu'à ce qu'il ait trouvé.
On pourra sortir du programme à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Il sera sans doute intéressant, pour de jeunes enfants, de prévoir une version non automatique où le joueur répondrait OUI ou NON pour chaque lettre proposée par le MO6 ; on pourra aussi imaginer d'ajouter un affichage graphique, par exemple une potence ou une voiture qui avance vers un précipice, etc.

```
10 REM PENDU INVERSE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
60 C=1:T$=""
70 INPUT"ENTREZ VOTRE MOT :";R$
80 CLS
90 FOR I=1 TO LEN(R$)
100 LOCATE I,5:PRINT"-"
110 T$=T$+"-"
120 NEXT I
130 LOCATE 0,9,0:PRINT"JE CHERCHE....; E
SSAI ";C
140 FOR J=1 TO 1000:NEXT J
150 X=-6*(RND<0.7)-13*(RND<0.5)-7*(RND<0
.4)
160 X=INT(X*RND+1)
170 IF MID$(D$,X,1)="-" THEN 150
180 E$=MID$(D$,X,1)
190 D$=LEFT$(D$,X-1)+"-"+RIGHT$(D$,26-X)
200 PRINT:PRINT:PRINT"JE PRENDS :";E$:PL
AY"O5;DO;RE;MI"
210 FOR I=1 TO LEN(R$)
220 IF E$=MID$(R$,I,1) THEN T$=LEFT$(T$,
I-1)+E$+RIGHT$(T$,LEN(T$)-I):PLAY"RE;RE;
SO"
230 LOCATE I,5,0:PRINT MID$(T$,I,1)
240 NEXT I
250 C=C+1
260 IF T$=R$ THEN LOCATE 0,18:PRINT"TROU
VE EN ";C-1;" COUPS":PLAY"DO;SO;MI":END
270 GOTO 130
```



Nicomaque

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion (calcul mental).
Difficulté du jeu : moyenne.

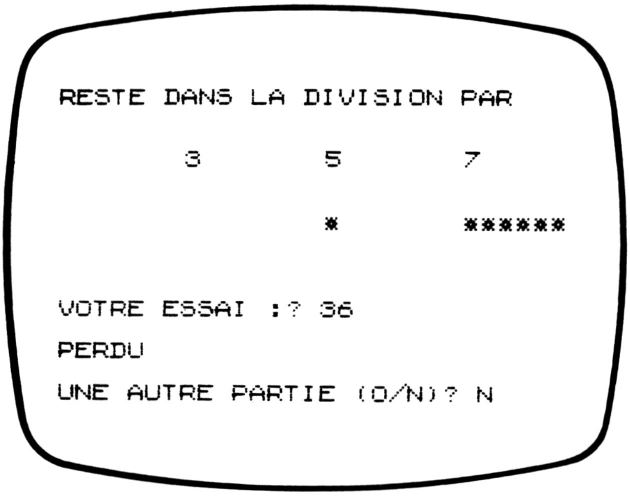
LE JEU :
Il utilise une propriété un peu particulière des entiers de 1 à 1ØØ qui peuvent être identifiés sans ambiguïté par leurs restes dans les divisions par 3, 5 et 7.
Le programme, proposé ici, est un programme codeur qui affiche simplement les restes d'un entier donné N dans les divisions indiquées.
Le jeu consiste à essayer de découvrir l'entier (mais, dans cette version, vous n'avez droit qu'à un seul essai).

LE PROGRAMME :
La ligne 5Ø permet de confectionner l'entier.
La boucle I (8Ø-13Ø) calcule les trois restes successifs, et la boucle emboîtée J (1ØØ-12Ø) permet de les afficher sous une forme assez visuelle, aux bons emplacements.
Les lignes 6Ø et 7Ø sont là pour indiquer de quelle division il s'agit.
Le nombre proposé par le joueur est entré à la ligne 14Ø ; il est affiché et testé aux lignes 15Ø et 16Ø (affichage de fin de partie : BRAVO ou PERDU).
Les lignes 17Ø et 18Ø s'occupent de gérer la prochaine partie, selon la réponse du joueur.

POUR JOUER :
On lance le programme en tapant RUN.
Les trois restes seront alors affichés sur la même ligne, symbolisés par des étoiles (l'absence d'étoiles signifiant : reste Ø).
A la question VOTRE ESSAI, on répondra en donnant un entier compris entre Ø et 1ØØ, suivi de ENTREE.
On obtiendra, en réponse, un affichage PERDU ou BRAVO, puis la question AUTRE PARTIE (O/N).
Pour faire un nouvel essai, on répondra O, suivi de ENTREE.

EXTENSIONS POSSIBLES :
On peut envisager de réaliser le programme décodeur (on donne les restes et le MO6 devra vous donner N) et chercher si la propriété peut être étendue (en effectuant davantage de divisions, par exemple).

```
10 REM NICOMAQUE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 X=INT(RND*100+1)
60 PRINT"RESTE DANS LA DIVISION PAR"
70 PRINT:PRINT TAB(12);"3";TAB(20);"5";T
AB(28);"7"
80 FOR I=1 TO 3
90 A=X-(2*I+1)*INT(X/(2*I+1))
100 FOR J=1 TO A
110 COLOR 3,1:LOCATE J+8*I+3,5:IF A<>0 THEN
PRINT "*"
120 NEXT J
130 NEXT I
140 COLOR 4,6:LOCATE 0,10:INPUT"VOTRE ES
SAI :";N
150 IF N=X THEN PRINT:PRINT"BRAVO"
160 IF N<>X THEN PRINT:PRINT"PERDU"
170 PLAY"DO;DO;DO":PRINT:INPUT"UNE AUTRE
 PARTIE (O/N)";R$
180 IF LEFT$(R$,1)="O" THEN 40
```



Course de chevaux

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : aucune.

LE JEU :
On entrera un nombre de chevaux compris entre 1 et 10, et la course débutera immédiatement.
Il pourra y avoir des ex aequo, car l'on tiendra compte du fait que le MO6 s'occupe de faire avancer les chevaux les uns après les autres, en commençant par le numéro 1 ; le classement sera affiché au fur et à mesure des arrivées (et en tenant compte des ex aequo).

LE PROGRAMME :
Une première boucle non formelle (80-170) affiche les chevaux dans leur position de départ.
Une deuxième boucle I (100-180) les fera avancer ; la position de chaque cheval sera repérée par un chiffre ou une lettre dans une variable alphanumérique A$.
La ligne 120 fera avancer les chevaux.
La ligne 140 gère les arrivées.
La ligne 150 réalisera les impressions.
Le sous-programme 240-280 fait avancer le compteur des arrivées, et imprime les résultats.

POUR JOUER :
On lance le programme par RUN.
A la question NOMBRE DE CHEVAUX : on répond par un entier de 1 à 10, suivi de ENTREE.
La course se déroulera ensuite automatiquement.

EXTENSIONS POSSIBLES :
On pourra dessiner les chevaux avec un graphisme plus soigné, et les différencier par leur couleur.
On pourra aussi imaginer tout un système de paris, qui interviendra avant le départ de chaque course.

```
10 REM COURSE DE CHEVAUX
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR J=1 TO F:Y=RND:NEXT J
40 COLOR 4,6:CLS
50 INPUT"NOMBRE DE CHEVAUX :";N:IF N>9 T
HEN 40
60 A$="":P=0:A=0:T=1:K=0
70 J=T
80 A$=A$+"0":I=J
90 GOTO 150
100 FOR I=1 TO N
110 IF MID$(A$,I,1)="K" THEN 180
120 B$=CHR$(ASC(MID$(A$,I,1))+INT(RND*4+
1))
130 A$=LEFT$(A$,I-1)+B$+RIGHT$(A$,LEN(A$
)-I)
140 IF B$>="K" THEN GOSUB 240
150 COLOR 4,6:LOCATE ASC(MID$(A$,I,1))-4
6,2*I+3,0:PRINT "     ";:COLOR 3,1:PRINT
I
160 PLAY"O5;SO":FOR D=1 TO 600:NEXT D
170 IF K=0 THEN J=J+1:IF J<=N THEN 80 EL
SE K=1:GOTO 100
180 NEXT I
190 P=P+T
200 T=0
210 IF A<N THEN 100
220 PLAY"DO;SO;MI"
230 COLOR 4,6:END
240 A$=LEFT$(A$,I-1)+"K"+RIGHT$(A$,LEN(A
$)-I)
250 COLOR 0,7:LOCATE 0,2*I+3:PRINT P
260 T=1:PLAY"O5;DO;DO;DO"
270 A=A+T
280 RETURN
```

A travers

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

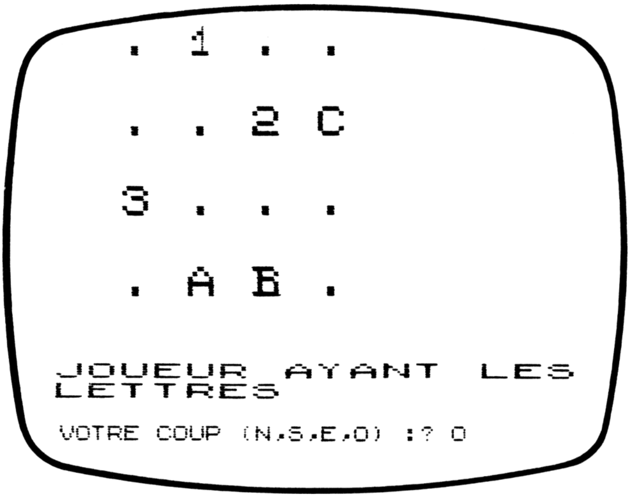
LE JEU :
Sur l'écran apparaît un carré de 9 cases, avec sur le bord vertical des lettres et sur le bord horizontal des chiffres.
L'un des joueurs manipule les chiffres, l'autre les lettres. 3 déplacements sont possibles pour les chiffres, et 3 autres pour les lettres. Les lettres peuvent se déplacer vers le haut, à droite et à gauche et les chiffres vers la droite, en haut ou en bas.
Le gagnant est le premier qui arrive à faire sortir tous ses symboles (les lettres sortent en haut, et les chiffres à droite).
Il n'y a pas de prise, et les sauts ne sont pas autorisés ; on ne peut pas non plus se placer sur une case déjà occupée.

LE PROGRAMME :
Les positions des chiffres et des lettres sont enregistrées dans une table N (5, 5) ; les lignes 1Ø à 7Ø définissent les valeurs de départ.
Les lignes 9Ø à 14Ø dessinent le terrain de jeu et la ligne 16Ø affiche le joueur qui a le trait.
La ligne 19Ø contrôle le coup joué, de même que les boucles I et J (2ØØ à 24Ø), et le changement de joueur est assuré par la ligne 33Ø.

POUR JOUER :
Le terrain de jeu sera affiché après la frappe de RUN et l'ordinateur attend le coup du joueur désigné.
On entrera d'abord le chiffre ou la lettre représentant le pion à déplacer, suivi d'une indication de direction sous la forme N (pour Nord), S (pour Sud), E (pour Est) ou O (pour Ouest).
On pourra ainsi faire 2E suivi de ENTREE.
Si le coup est invalide, il ne sera pas exécuté, et le code du joueur restera à l'affichage ; s'il est autorisé, le déplacement sera effectué, et l'on obtient l'affichage correspondant au deuxième joueur.
Le premier qui aura réussi à faire disparaître ses 3 pièces aura gagné.

EXTENSIONS POSSIBLES :
On peut envisager d'étendre le jeu avec davantage de chiffres ou de lettres et d'améliorer les contrôles. On peut aussi définir une stratégie pour le MO6.

```
10 REM A TRAVERS
20 DIM N(5,5):K=1
30 FOR I=1 TO 4
40 N(I,1)=ASC("0")+I
50 N(4,I)=ASC("?")+I
60 NEXT I
70 N(4,1)=0
80 CLS : SCREEN 4,6,5
90 FOR U=1 TO 4
100 FOR V=1 TO 4
110 COLOR 1,6:ATTRB 1,1
120 LOCATE 4*U,4*V-2:IF N(V,U)=0 THEN PR
INT "." ELSE PRINT CHR$(N(V,U))
130 NEXT V
140 NEXT U
150 COLOR 0,7:ATTRB 1,0:LOCATE 0,18
160 PRINT"JOUEUR AYANT LES ";:IF K=0 THE
N PRINT"LETTRES" ELSE PRINT"CHIFFRES"
170 COLOR 4,6:ATTRB 0,0:LOCATE 0,22
180 INPUT"VOTRE COUP (N,S,E,O) :";R$
190 IF (K=1 AND RIGHT$(R$,1)="O") OR (K=
0 AND RIGHT$(R$,1)="S") THEN 170
200 FOR I=1 TO 4
210 FOR J=1 TO 4
220 IF N(I,J)=ASC(LEFT$(R$,1)) THEN 260
230 NEXT J
240 NEXT I
250 GOTO 150
260 A=I:B=J
270 I=I-(RIGHT$(R$,1)="S")+(RIGHT$(R$,1)
="N")
280 J=J-(RIGHT$(R$,1)="E")+(RIGHT$(R$,1)
="O")
290 IF (I=0 AND K=0) OR (J>4 AND K=1) TH
EN 320
300 IF N(I,J)<>0 THEN 150
310 N(I,J)=ASC(LEFT$(R$,1))
320 N(A,B)=0
330 K=ABS(K-1)
340 GOTO 80
```



Mastermind

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : logique (un seul joueur).
Difficulté du jeu : difficile.

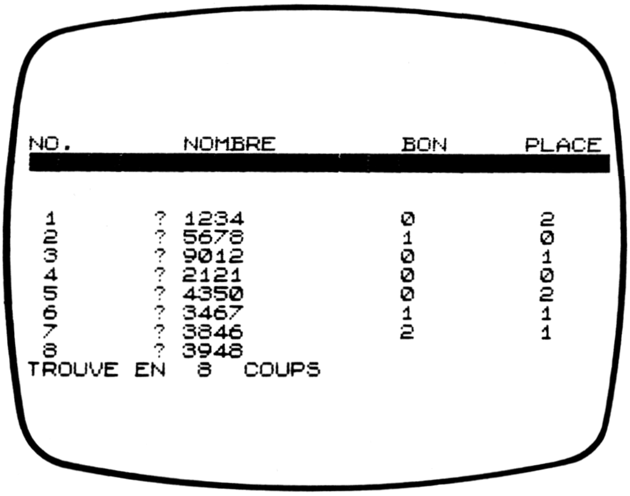
LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, ce jeu développe le raisonnement déductif ; il faut réussir à trouver le plus vite possible un nombre de n chiffres (on pourra prendre n compris entre 1 et 5), en disposant à chaque essai de renseignements.
En effet, le MO6 répondra en indiquant d'une part le nombre de chiffres qui sont, dans le nombre proposé, exactement à la même place que dans le nombre à découvrir, et, d'autre part, le nombre de chiffres qui figurent dans le nombre à découvrir sans être exactement au bon emplacement.

LE PROGRAMME :
Le choix du nombre de chiffres du nombre à découvrir est effectué aux lignes 4Ø et 5Ø.
Le nombre est fabriqué à la ligne 6Ø (il pourra y avoir des chiffres répétés).
L'affichage est préparé par la ligne 9Ø.
La ligne 13Ø s'occupe de la fin de partie.
Une première boucle I (14Ø-23Ø) teste les chiffres qui sont à la bonne place.
Une deuxième boucle J (16Ø-28Ø) teste les chiffres qui sont présents et mal placés.
Les compteurs sont aux lignes 2ØØ et 22Ø.
La ligne 24Ø affiche le résultat, et la ligne 25Ø comptabilise les essais.

POUR JOUER :
On tape RUN et l'on choisit la longueur du nombre désiré (de 1 à 6), suivie de ENTREE.
Après l'affichage NO. NOMBRE BON PLACE, on introduit son propre essai, suivi de ENTREE.
On obtient immédiatement la réponse (en cas de double chiffre, les résultats peuvent être un peu délicats à interpréter).
On introduit ensuite le nouvel essai, et ainsi de suite.
Lorsque l'on a trouvé, on obtiendra l'affichage TROUVE EN ... COUPS.
On peut sortir du programme à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut chercher à éviter la présence de chiffres doubles dans le nombre à trouver, on peut aussi prévoir un changement de page automatique, avec rappel des résultats précédents ; on pourra aussi réaliser des jeux analogues avec des mots, ou d'autres symboles (couleur, taille...).

```
10 REM MASTERMIND
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 INPUT"COMBIEN DE CHIFFRES :";N
50 IF N>6 THEN 40
60 E=1:F=1:X$=STR$(INT(RND*9^N+10^(N-1))
)
70 X$=RIGHT$(X$,LEN(X$)-1)
80 CLS
90 PRINT"NO.          NOMBRE          BON
 PLACE"
100 COLOR 1:FOR J=1 TO 40:PRINT CHR$(127
);:NEXT J
110 COLOR 4,6:LOCATE 0,3+F:PRINT E;TAB(8
);:INPUT E$
120 X=0:Y=X
130 IF E$=X$ THEN PLAY"DO;SO;MI":PRINT"T
ROUVE EN ";E;" COUPS":END
140 FOR I=1 TO N
150 IF MID$(X$,I,1)=MID$(E$,I,1) THEN 22
0
160 FOR J=1 TO N
170 IF J<>I AND MID$(X$,I,1)=MID$(E$,J,1
) THEN 200
180 NEXT J
190 GOTO 230
200 Y=Y+1
210 GOTO 230
220 X=X+1
230 NEXT I
240 LOCATE 23,3+F:PRINT X;TAB(32);Y:PLAY
"DO;RE;MI"
250 E=E+1:F=F+1:IF F=18 THEN F=1
260 GOTO 110
```



Idem

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de calcul mental (2 joueurs).
Difficulté du jeu : selon le niveau choisi.

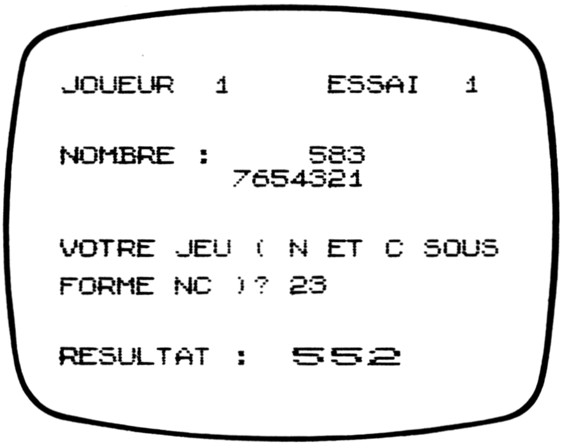
LE JEU :
Le MO6 choisit un nombre de n chiffres (selon le niveau de difficulté indiqué au début). Ce nombre (le même au début pour les 2 joueurs) sera régulièrement décrémenté à chaque tour, d'une quantité également choisie au hasard par le MO6 en début de jeu ; vous devez, à chaque coup, choisir une colonne (les dizaines, les centaines, etc.) et ajouter dans cette colonne un certain entier ; le but est d'arriver à obtenir à un moment l'affichage d'un nombre ayant tous ses chiffres identiques.
Chaque résultat sera affiché un court instant, après quoi la main passera au joueur suivant, qui repartira de son précédent résultat.

LE PROGRAMME :
Une variable numérique N est dimensionnée à 2 en ligne 4Ø, pour contenir les résultats successifs des deux joueurs.
La ligne 5Ø permettra d'entrer le niveau de difficulté choisi.
Le nombre initial est confectionné par la ligne 7Ø. La quantité qui interviendra dans les décrémentations est obtenue en ligne 9Ø.
La ligne 12Ø teste si le résultat n'est pas trop petit.
La ligne 13Ø affiche le numéro du joueur.
Le nombre est affiché en ligne 14Ø et le jeu du joueur est entré en ligne 16Ø (nombre de deux chiffres).
La ligne 17Ø incrémente la colonne choisie, et la ligne 18Ø décrémente le nombre de la quantité choisie au départ.
Le résultat sera affiché un court instant (ligne 21Ø). La boucle I (26Ø-28Ø) permet de vérifier si tous les chiffres sont identiques.
L'impression et la fin de partie sont effectuées en ligne 29Ø. Les lignes 3ØØ et 31Ø s'occupent du changement de joueur.

POUR JOUER :
Lancer l'exécution du programme par RUN.
Le nombre initial sera affiché pour le joueur n° 1, qui joue en tapant xy, suivi de ENTREE ; x est le chiffre à ajouter, et y est le numéro de la colonne où il faut l'ajouter.
Le résultat (provisoire s'il est trop petit) sera alors affiché un court instant, et le trait passe au deuxième joueur, et ainsi de suite. On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra étendre le jeu à plus de 2 joueurs, améliorer le choix du nombre intervenant dans la décrémentation, améliorer les divers contrôles, laisser les différents résultats affichés, etc.

```
10 REM IDEM
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:H=1:DIM N(2)
50 INPUT"NIVEAU (2-7) :";N
60 U=10:P=0
70 N(1)=INT((U^N-U^(N-1))*RND+U^(N-1))
80 N(2)=N(1)
90 A=INT(2*U^(N-1)*RND+U^(N-1))
100 IF H=1 THEN P=P+1
110 CLS
120 IF N(H)<U^(N-1) THEN N(H)=U^N-ABS(N(
H))
130 LOCATE 0,3:PRINT"JOUEUR ";H;"     ESS
AI ";P
140 LOCATE 0,6:PRINT"NOMBRE :":LOCATE 15
-N,6:COLOR 3,1:PRINT N(H):COLOR 4,6
150 LOCATE 9,7:COLOR 3,0:PRINT"7654321 "
:COLOR 4,6
160 LOCATE 0,10:INPUT"VOTRE JEU ( N ET C
 SOUS FORME NC )";R$
170 N(H)=N(H)+VAL(LEFT$(R$,1))*U^(VAL(RI
GHT$(R$,1))-1)
180 N(H)=N(H)-A
190 IF N(H)>U^N THEN N(H)=N(H)-U^N
200 N(H)=INT(N(H))
210 LOCATE 0,13:PRINT"RESULTAT :";:COLOR
 5,6:ATTRB 1,0:PRINT N(H)
220 COLOR 4,6:ATTRB 0,0:PLAY"DO;RE;MI"
230 FOR J=1 TO 1200:NEXT J
240 R$=STR$(N(H))
250 R$=RIGHT$(R$,LEN(R$)-1)
260 FOR I=1 TO LEN(R$)-1
270 IF MID$(R$,I,1)<>MID$(R$,I+1,1) THEN
 300
280 NEXT I
290 PRINT:PRINT"BRAVO , JOUEUR ";H:PLAY"
DO;SO;MI":END
300 H=H+1
310 IF H>=3 THEN H=1
320 GOTO 100
```



Horloge 1

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

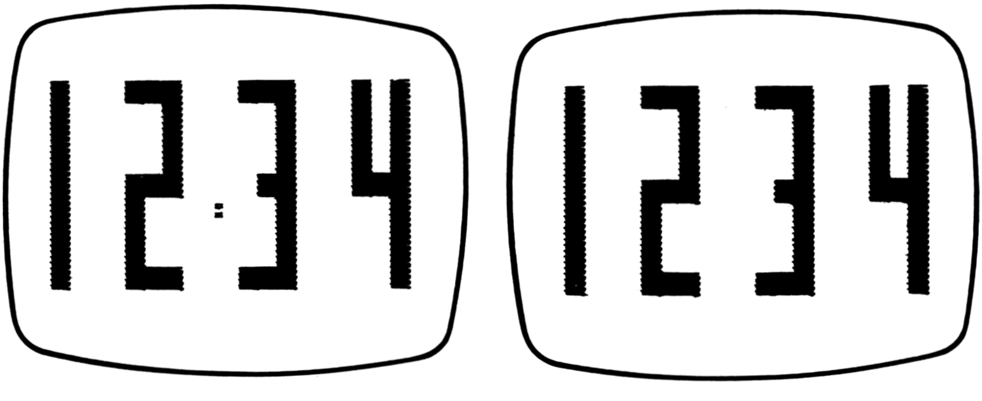
LE JEU :
Transformer votre écran de télévision en horloge numérique à affichage grand format, grâce à ce programme qui vous permettra d'animer une vitrine, de faire une horloge fantaisie, et qui pourra se prêter à bien d'autres usages.

LE PROGRAMME :
La ligne 3Ø vous permettra d'entrer l'heure sous forme d'un entier de 4 chiffres (par exemple : 1Ø24 signifera 1Ø heures 24 minutes).
Les lignes 8Ø-17Ø réalisent le dessin, en décalant l'impression au fur et à mesure (à l'aide de la variable K, incrémentée en ligne 16Ø).
Les lignes 18Ø à 22Ø assurent la progression de l'heure.
Les lignes 23Ø à 29Ø effectuent l'affichage de deux points clignotants pour les secondes.

POUR JOUER :
Dès le lancement par RUN, le programme se place en attente de l'heure ; on tape et on attend le top pour appuyer sur ENTREE.
L'heure s'affiche alors assez rapidement et l'horloge est en route. Pour l'arrêter, il faudra appuyer sur les touches CNT et C.
L'heure sera entrée sous forme d'un entier ayant impérativement 4 chiffres, pour 5 heures 6 minutes, on veillera à taper Ø5Ø6 suivi de ENTREE.

EXTENSIONS POSSIBLES :
Il sera intéressant de modifier le programme pour en faire un réveil, avec alarme ; on peut aussi améliorer le graphisme, introduire la couleur.

```
10 REM HORLOGE 1
20 CLS
30 INPUT"HEURE (FORME XXXX )";A$
40 CLS:SCREEN 1,4,0
50 K=1
60 A=VAL(MID$(A$,K,1))
70 IF A <>1 AND A<>4 THEN LOCATE 6*K-5,1
:PRINT CHR$(127)+CHR$(127)+CHR$(127)
80 FOR J=1 TO 5
90 IF A=0 OR (A>3 AND A<>7) THEN LOCATE
6*K-5,J:PRINT CHR$(127)
100 IF A<>5 AND A<>6 THEN LOCATE 6*K-3,J
:PRINT CHR$(127)
110 IF A=0 OR A=2 OR A=6 OR A=8 THEN LOC
ATE 6*K-5,J+4:PRINT CHR$(127)
120 IF A<>2 THEN LOCATE 6*K-3,J+4:PRINT
CHR$(127)
130 NEXT J
140 IF A>1 THEN LOCATE 6*K-4,5:PRINT CHR
$(127)+CHR$(127)
150 IF A<>1 AND A<>4 AND A<>7 THEN LOCAT
E 6*K-5,9:PRINT CHR$(127)+CHR$(127)+CHR$
(127)
160 K=K+1
170 IF K<5 THEN 60
180 A=1+VAL(A$)
190 IF A=2360 THEN A=0
200 A$="000"+RIGHT$(STR$(A),LEN(STR$(A))
-1)
210 A$=RIGHT$(A$,4)
220 IF RIGHT$(A$,2)="60" THEN A$=RIGHT$(
STR$(VAL(LEFT$(A$,2))+1),2)+"00"
230 H=1
240 FOR J=1 TO 128
250 FOR I=1 TO 24
260 LOCATE 11,6,0:ATTRB 1,0:IF H=1 THEN
PRINT ":" ELSE PRINT " ":ATTRB 0,0
270 NEXT I
280 H=ABS(H-1)
290 NEXT J
300 PLAY"DO;SI;FA":GOTO 40
```



Reines sur l'échiquier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

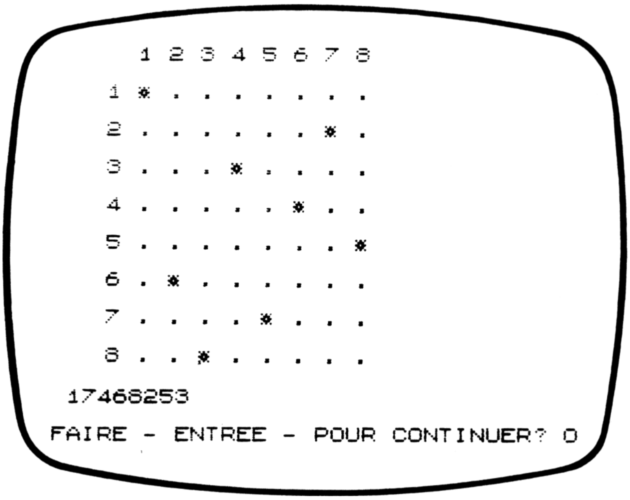
LE JEU :
Le programme utilise un algorithme très performant pour résoudre un problème qui a donné du fil à retordre à des chercheurs, même aussi célèbres que le mathématicien GAUSS, qui n'avait pas pu découvrir toutes les solutions. Il s'agit de trouver toutes les manières de disposer des reines sur un échiquier, de telle sorte qu'aucune ne puisse en prendre une autre. Le programme proposé est très souple, puisque vous disposez même de la possibilité de choisir la taille de votre échiquier, et les résultats seront affichés à la fois sous forme visuelle et sous forme numérique.

LE PROGRAMME :
La ligne 3Ø vous permet de choisir la taille de l'échiquier.
Une variable D sera dimensionnée en conséquence, en ligne 4Ø.
Une boucle d'exploration principale I (5Ø-11Ø) contient une sous-boucle J (8Ø-1ØØ) qui réalise le contrôle de non-alignement (ligne 9Ø).
Si, au cours de l'exploration, on rencontre un alignement, on revient en arrière, en utilisant les lignes 23Ø à 27Ø et on recommence.
La boucle F (13Ø-21Ø) assure l'impression de l'échiquier, et la boucle K (16Ø-18Ø), que l'on pourra éventuellement supprimer, assure l'impression du résultat sous forme d'un code numérique correspondant à la position des pièces dans chaque colonne.
La ligne 22Ø permet un arrêt après l'affichage de chacune des solutions.

POUR JOUER :
On lance le programme avec RUN, puis on tape la dimension de l'échiquier désirée, et on fait ENTREE.
Il suffit d'attendre, avec patience (plus l'échiquier sera grand, plus l'attente sera longue). La première solution sera affichée puis le programme s'arrête, et il faut appuyer sur la touche ENTREE pour le faire repartir. Pour terminer l'exploration sans avoir eu toutes les solutions, il faudra taper CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager de trouver des algorithmes pour tous les problèmes analogues (position de tours, déplacement d'un cavalier, etc.).

```
10 REM REINES SUR L'ECHIQUIER
20 CLS
30 INPUT"TAILLE DE L'ECHIQUIER :";S
40 DIM D(S)
50 I=1
60 D(I)=1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF D(I)=D(J) OR (ABS(D(I)-D(J))=I-J)
THEN 240
100 NEXT J
110 I=I+1:IF I<=S THEN 60
120 CLS
130 FOR F=1 TO S
140 LOCATE 2*F,0:PRINT F
150 LOCATE 0,2*F:PRINT F
160 FOR K=1 TO S
170 LOCATE 2*K+1,2*F:PRINT "."
180 NEXT K
190 LOCATE 2*D(F)+1,2*F:COLOR 3,1:PRINT
"*"
200 LOCATE F,20:COLOR 4,6:PRINT RIGHT$(S
TR$(D(F)),1)
210 NEXT F
220 PLAY"DO;SO;MI":LOCATE 0,22:INPUT"FAI
RE - ENTREE - POUR CONTINUER";R$
230 I=I-1
240 D(I)=D(I)+1
250 IF D(I)<=S THEN 70
260 I=I-1
270 IF I<>0 THEN 240
```



Répétitions

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de mémoire (1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

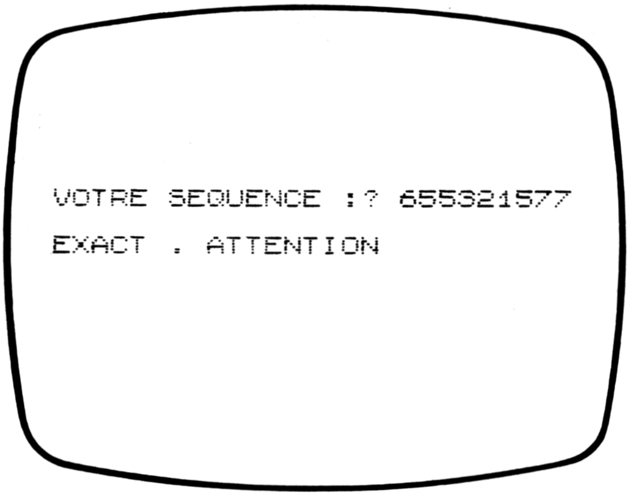
LE JEU :
Des nombres vont apparaître un par un.
Le joueur devra à chaque coup les répéter tous, depuis le début (ce qui signifie qu'il en aura un de plus à taper à chaque fois).
Il devra essayer de tenir le plus longtemps possible. Le meilleur score est mémorisé après chaque partie, ce qui permettra éventuellement de jouer à plusieurs.

LE PROGRAMME :
La ligne 6Ø appelle un sous-programme (lignes 1ØØØ à 1Ø6Ø) qui donne la règle du jeu.
La ligne 8Ø fabrique les chiffres, qui sont ensuite stockés dans une variable alphanumérique A$.
La ligne 9Ø permet l'impression du dernier chiffre au hasard sur la quatrième ligne de l'écran, et la durée de l'affichage est déterminée par la ligne 1ØØ.
La ligne 12Ø attend la réponse du joueur, qui est comparée à la réponse correcte en ligne 13Ø.
La ligne 16Ø marque les points.
La ligne 17Ø contrôle le meilleur score.
Les affichages de fin de partie sont effectués de la ligne 2ØØ à la ligne 24Ø.

POUR JOUER :
Le lancement du programme par RUN provoque l'affichage de la règle du jeu, puis d'un chiffre qui disparaît. L'appareil pose alors la question VOTRE SEQUENCE.
On lui répond en redonnant le chiffre, suivi de ENTREE. Si c'est exact, on a un court instant l'affichage de EXACT. ATTENTION, puis un nouveau chiffre est affiché, qui s'efface, et de nouveau la question VOTRE SEQUENCE. On retape alors les deux chiffres déjà obtenus, suivis de ENTREE.
On continue ainsi (3 chiffres, puis 4, 5, etc.) jusqu'à ce que l'on fasse une erreur, ce qui provoque l'affichage du score (et du meilleur score obtenu jusqu'à présent), suivi de la question AUTRE PARTIE (O/N) ; on tape O pour continuer.
On peut aussi quitter le programme à tout moment, en tapant CNT et C avant l'affichage d'un chiffre.

EXTENSIONS POSSIBLES :
On pourra réaliser des jeux analogues et plus ou moins perfectionnés en remplaçant les chiffres par des lettres ou des couleurs ou des sons...

```
10 REM REPETITIONS
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 M=0
50 A$=""
60 GOSUB 1000
70 H=0
80 X=INT(RND*10):A$=A$+RIGHT$(STR$(X),1)
90 CLS:ATTRB 1,1:COLOR 1,6:LOCATE RND*40
,4:PRINT X
100 FOR I=1 TO 200+RND*300:NEXT I
110 CLS:ATTRB 0,0:COLOR 4,6
120 LOCATE 0,10:INPUT"VOTRE SEQUENCE :";
R$
130 IF R$<>A$ THEN 190
140 PRINT:PLAY"DO;RE;MI":PRINT"EXACT . A
TTENTION"
150 FOR I=1 TO 400:NEXT I
160 H=H+1
170 IF M<H THEN M=H
180 GOTO 80
190 CLS
200 PRINT"VOTRE SCORE EST DE ";H
210 PRINT:PRINT"MEILLEUR SCORE ACTUEL "M
220 PRINT:INPUT"AUTRE PARTIE (O/N)";R$
230 IF R$="N" THEN PLAY"DO;DO;DO":END
240 GOTO 50
1000 CLS
1010 PRINT"VOUS DEVEZ REPETER TOUTE LA S
ERIE DES"
1020 PRINT"NOMBRES APPARUS DEPUIS LE DEB
UT DU JEU"
1030 PRINT:PRINT"IL Y EN A UN DE PLUS A
CHAQUE FOIS"
1040 PRINT:PRINT"APPUYER SUR - ENTREE -
POUR COMMENCER"
1050 INPUT R$
1060 RETURN
```



Mots

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : jeu de lettres (pour 1 ou plusieurs joueurs).
Difficulté du jeu : moyenne.

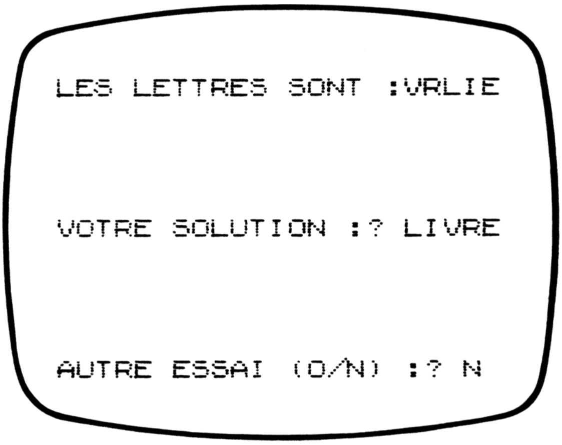
LE JEU :
Il est difficile d'imaginer plus simple dans son principe que ce jeu, qui se contente d'afficher dans le désordre les lettres prises dans une trentaine de mots. Mais essayez de jouer, et vous verrez que ce n'est pas si facile, la difficulté variant évidemment selon la liste de mots que vous aurez introduits en DATA au préalable.

LE PROGRAMME :
La ligne 4Ø permet de dimensionner la variable A$ (I) qui contiendra les mots.
La boucle I (5Ø) saisit la liste des mots (inscrits en DATA aux lignes 24Ø à 28Ø).
La ligne 7Ø choisit un mot au hasard ; la boucle I (1ØØ-15Ø) contrôle si une lettre n'a pas déjà été tirée, et reprend les lettres dans le désordre.
La ligne 16Ø affiche les lettres, et la ligne 17Ø attend votre proposition, qui est testée en ligne 18Ø.
Le nombre d'essais est incrémenté en ligne 19Ø.
La fin de partie est contrôlée par la ligne 2ØØ, et le score est affiché en ligne 22Ø.

POUR JOUER :
On lance le programme par RUN. Une série de lettres est affichée dans le désordre, et il faut essayer de retrouver le mot correspondant.
On tape le mot supposé, puis ENTREE, et une nouvelle série de lettres sera affichée, si l'on répond O à la question AUTRE ESSAI (O/N).
Le score et le nombre d'essais sont affichés en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer de minuter le temps d'apparition du mot, ou de faire défiler les lettres l'une après l'autre sur l'écran.
On peut aussi corser la difficulté en introduisant des JOKERS, jouer à plusieurs, tenir le compte des points, etc.

```
10 REM MOTS
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 DIM A$(30):S=0:E=0
50 FOR I=1 TO 30:READ A$(I):NEXT I
60 CLS
70 Z=INT(RND*30+1)
80 IF A$(Z)="" THEN 70
90 R$=A$(Z):A$="":A$(Z)="":V$=R$
100 FOR I=1 TO LEN(R$)
110 Z=INT(RND*LEN(R$)+1)
120 IF MID$(R$,Z,1)="-" THEN 110
130 A$=A$+MID$(R$,Z,1)
140 R$=LEFT$(R$,Z-1)+"-"+RIGHT$(R$,LEN(R
$)-Z)
150 NEXT I
160 PRINT"LES LETTRES SONT :";A$:PLAY"DO
;RE;MI"
170 LOCATE 0,6:INPUT"VOTRE SOLUTION :";S
$
180 IF S$=V$ THEN S=S+1
190 E=E+1
200 LOCATE 0,12:INPUT"AUTRE ESSAI (O/N)
:";R$
210 IF LEFT$(R$,1)="O" THEN 60
220 LOCATE 0,15:PRINT"VOTRE SCORE :";S;"
 SUR ";E;" ESSAIS"
230 PLAY"DO;SO;MI":END
240 DATA JAMBON,ENERGUMENE,AFFILIE,ERUDI
T,INSIGNIFIANT,REALISTE
250 DATA IMPRESSIONNANT,ALLECHANT,EPIQUE
,INTREPIDE,COUARD,ORDINATEUR
260 DATA TREVE,BANDEROLLE,SENAT,ACTUEL,C
ONSACRE,DIEDRE
270 DATA LIVRE,TERMINE,SIGNATURE,MALODOR
ANT,AIMANT,ARBITRAIRE
280 DATA CHALEUR,OPPORTUN,PELURE,ALTRUIS
TE,RAMONEUR,NOVICE
```



Permutations

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion.
Difficulté du jeu : moyenne.

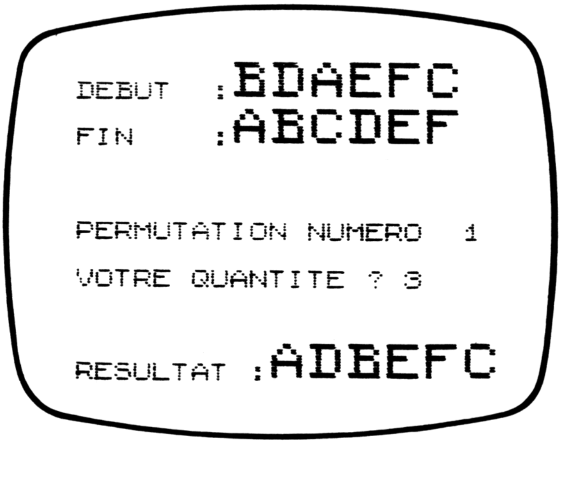
LE JEU :
Le MO6 vous demande d'abord le nombre de lettres que vous désirez permuter (pas plus de 12). A partir de là, il va vous afficher le nombre correspondant de lettres prises dans le début de l'alphabet, d'abord dans le désordre (DEBUT) puis dans l'ordre (FIN).
Vous devez réussir à remettre dans l'ordre les lettres données au début, en utilisant des permutations sur les n premières lettres à partir de la gauche, avec le moins d'essais possible.

LE PROGRAMME :
La ligne 40 permet de choisir le nombre de lettres, et la ligne 50 contrôle que l'on n'en demande pas plus de 12.
La ligne 80 réalise le brouillage, en tirant au hasard la nouvelle position de chaque lettre. Les lettres sont ensuite ajoutées au fur et à mesure de leur tirage, dans une variable B$ (ligne 100), supprimées dans A$ et replacées en C$, dans le bon ordre (lignes 110-120).
La ligne 140 contrôle le nombre de lettres de B$.
Les affichages sont effectués aux lignes 160, 170 et 180.
Le retournement est fait, lettre par lettre, dans la boucle I (210-230) ; le résultat est placé dans B$ en ligne 240, affiché par la ligne 250 et comparé au résultat final en ligne 260.
Le nombre d'essais sera affiché en fin de partie, grâce à la ligne 270.

POUR JOUER :
Après avoir tapé RUN, on obtient l'affichage de la question NOMBRE DE LETTRES (DE 1 A 12) ; on répond en frappant N, suivi de ENTREE.
L'affichage suivant présente l'ordre initial (DEBUT) et l'ordre à obtenir (FIN).
On joue en indiquant le nombre de lettres à permuter à partir de la gauche, par exemple : sur un affichage initial de GEABDFC, si l'on tape 3 et ENTREE, on obtiendra AEGBDFC.
On poursuivra jusqu'à l'obtention de l'ordre désiré.
Le score sera affiché en fin de partie ainsi qu'une question AUTRE PARTIE (O/N) ; répondre O si l'on désire faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra utiliser le même type de jeu pour, par exemple, reconstituer un dessin caché dont les diverses parties seraient repérées par des lettres. On peut aussi réaliser des permutations plus complexes (par exemple le fait de permuter deux lettres permute les deux voisines de la lettre la plus à gauche, etc.).

```
10 REM PERMUTATIONS
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 PRINT:INPUT"NOMBRE DE LETTRES ( DE 1
A 12 )";N
50 IF N>12 THEN 40
60 A$="ABCDEFGHIJKL":C$="            ":B
$=""
70 CLS
80 X=INT(RND*N+1)
90 IF MID$(A$,X,1)=" " THEN 80
100 B$=MID$(A$,X,1)+B$
110 C$=LEFT$(C$,X-1)+MID$(A$,X,1)+RIGHT$
(C$,12-X)
120 A$=LEFT$(A$,X-1)+" "+RIGHT$(A$,12-X)
130 S=1
140 IF LEN(B$)<>N THEN 80
150 C$=LEFT$(C$,LEN(B$))
160 LOCATE 0,4:PRINT"DEBUT  :";:COLOR 1,
6:ATTRB 1,1:PRINT B$:ATTRB 0,0:COLOR 4,6
170 LOCATE 0,6:PRINT"FIN    :";:COLOR 0,
6:ATTRB 1,1:PRINT C$:ATTRB 0,0:COLOR 4,6
180 LOCATE 0,10:PRINT"PERMUTATION NUMERO
 ";S:PRINT:INPUT"VOTRE QUANTITE ";L:IF L
>N THEN 180
190 S=S+1
200 A$=""
210 FOR I=1 TO L
220 A$=A$+MID$(B$,L-I+1,1)
230 NEXT I
240 B$=A$+RIGHT$(B$,LEN(B$)-L)
250 PLAY"DO;RE;MI":LOCATE 0,16:PRINT"RES
ULTAT :";:COLOR 5,6:ATTRB 1,1:PRINT B$:C
OLOR 4,6:ATTRB 0,0
260 IF B$<>C$ THEN 180
270 PLAY"DO;SO;MI":PRINT:PRINT"TROUVE EN
 ";S-1;" ESSAIS"
280 PRINT:INPUT"AUTRE PARTIE ( O/N )";R$
290 IF LEFT$(R$,1)="O" THEN 40
```



Bridge

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : variable, selon le problème posé.

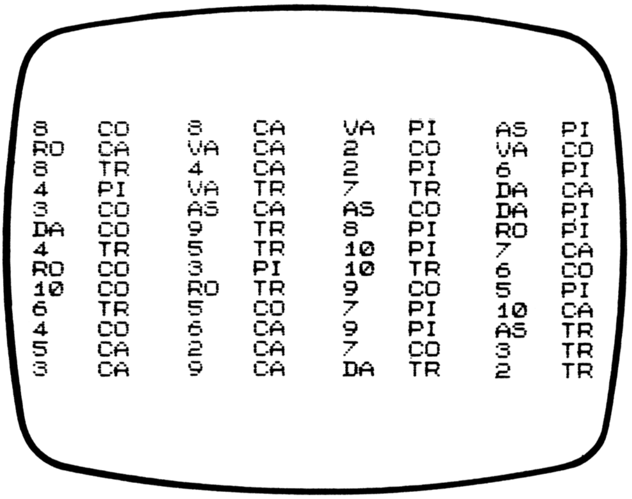
LE JEU :
Vous ne trouverez pas, dans ce programme, un partenaire pour vos bridges mais, plus exactement, une simulation pour l'affichage d'une donne ; cet affichage est réalisé jeu par jeu. On peut, à partir de là, se poser toutes sortes de problèmes : annonces, jeu de la carte, etc.

LE PROGRAMME :
Les lignes 50 et 60 définissent un jeu de 52 cartes.
La variable B$, en ligne 70, permettra de donner en abrégé le nom des couleurs et aussi d'améliorer la désignation de certaines cartes (10, valet, dame, roi et as).
La boucle J (80-220) réalise le tirage de quatre jeux de 13 cartes.
La boucle I (90-190) réalise tirage et impression. La carte est choisie parmi les 52 disponibles, en ligne 100.
La ligne 110 calcule la couleur de la carte tirée.
La ligne 120 vérifie que cette carte n'a pas déjà été tirée.
Les lignes 130 et 140 s'occupent de l'impression complémentaire dans le cas où l'on tire un 10, un valet, une dame, un roi ou un as.
L'impression est réalisée par les lignes 150, 160 et 170.
La ligne 180 permet d'éliminer les cartes déjà tirées, et la ligne 200 réalise un arrêt entre chaque affichage.

POUR JOUER :
On lance le programme par RUN.
Une première série de 13 cartes est affichée verticalement. Il suffira d'appuyer sur la touche ENTREE pour obtenir la deuxième série de 13 cartes. On procédera de même pour la troisième et la quatrième.

EXTENSIONS POSSIBLES :
On peut imaginer de représenter les 13 cartes sous une forme plus conventionnelle.
On peut aussi ajouter la possibilité de faire des annonces (et de les comparer à celles du MO6, par exemple), ou introduire des possibilités de jeu de la carte, y compris pour le MO6.

```
10 REM BRIDGE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 0,1
50 R$="234567891VDRA"
60 R$=R$+R$+R$+R$
70 B$="TRCACOPI0AAOS"
80 FOR J=1 TO 4
90 FOR I=1 TO 13
100 X=INT(RND*LEN(R$)+1)
110 S=2*INT((X-1)/13)+1
120 IF MID$(R$,X,1)="0" THEN 100
130 T=X-13*INT(X/13)
140 T=T-13*(T=0)
150 LOCATE 10*J-10,I:COLOR 0,1:PRINT MID
$(R$,X,1);
160 IF T>=9 THEN PRINT MID$(B$,T,1)
170 LOCATE 10*J-6,I:PRINT MID$(B$,S,2)
180 R$=LEFT$(R$,X-1)+"0"+RIGHT$(R$,LEN(R
$)-X)
190 NEXT I
200 LOCATE 0,18:COLOR 4,6:INPUT"FAIRE -
ENTREE - POUR CONTINUER";S$
210 LOCATE 0,18:PRINT"
                                "
220 NEXT J
```



Hanoï

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

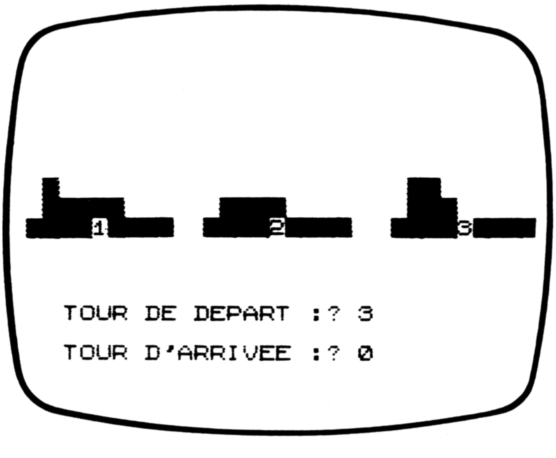
LE JEU :
Il est présenté, ici, dans une version pour un joueur. On dispose d'un certain nombre de disques (8 au plus) et l'on doit les transférer de la tour numéro 1 à la tour numéro 3, en respectant une seule règle : à aucun moment, un disque ne peut être recouvert par un disque de diamètre supérieur.
Le nombre de coups n'est pas comptabilisé par ce programme mais, par contre, les coups illicites seront refusés.

LE PROGRAMME :
La variable D$, en ligne 4Ø, servira à dessiner les disques.
Le nombre de disques est entré en ligne 5Ø et refusé s'il y a plus de 8 disques.
La boucle I (7Ø-9Ø) permet de fabriquer une variable A (1) qui contient l'état de la première pile sous forme numérique (en ligne 3Ø, trois variables sont prévues pour conserver l'état des trois piles).
Les socles sont dessinés par la ligne 11Ø et les trois piles par les boucles emboîtées I (13Ø-18Ø) et J (14Ø-17Ø).
Le contrôle du coup est effectué par la ligne 26Ø et l'état des piles est modifié par les lignes 27Ø et 28Ø.

POUR JOUER :
Après un RUN qui lance le programme, on entre le nombre de disques demandé, suivi de ENTREE. On obtient le dessin de l'état initial et, à la question TOUR DE DEPART, on répond en donnant le numéro de la tour de départ. Ce numéro (1, 2 ou 3) sera refusé si la tour est vide. Puis on tape ENTREE et, à la question TOUR D'ARRIVEE, on répond par le numéro de la tour d'arrivée.
Si le coup n'est pas possible, on sera obligé de recommencer à TOUR DE DEPART. Si le coup est possible, le déplacement est réalisé et l'on recommence.

EXTENSIONS POSSIBLES :
On pourra dessiner des disques de différentes couleurs, introduire un compteur de coups et, surtout, définir un jeu pour l'ordinateur.
On peut aussi essayer de compliquer le jeu avec deux tiges de plus et deux joueurs, chacun ayant des disques de couleur différente, par exemple.

```
10 REM HANOI
20 CLS
30 DIM A$(3):A$(1)="9":A$(2)="9":A$(3)="
9"
40 D$="":FOR I=1 TO 9:D$=D$+CHR$(127):NE
XT I
50 INPUT"NOMBRE DE TOURS (MOINS DE 8 ):"
;N
60 IF N>=8 THEN 50
70 FOR I=0 TO N-1
80 A$(1)=A$(1)+RIGHT$(STR$(N-I),1)
90 NEXT I
100 CLS
110 LOCATE 0,14:COLOR 5,6:PRINT D$:LOCAT
E 12,14:PRINTD$:LOCATE 24,14:PRINTD$
120 LOCATE 4,14:COLOR 0,5:PRINT"1":LOCAT
E 16,14:PRINT"2":LOCATE 28,14:PRINT"3"
130 FOR I=0 TO N-1
140 FOR J=1 TO 3
150 B$=A$(J)
160 IF LEN(B$)>I+1 THEN LOCATE 12*J-11,1
3-I:COLOR 1,6:PRINT LEFT$(D$,VAL(MID$(B$
,I+2,1)))
170 NEXT J
180 NEXT I
190 COLOR 4,6
200 LOCATE 0,16:INPUT"TOUR DE DEPART :";
D
210 IF D>3 OR D<1 THEN 200
220 IF A$(D)="9" THEN 200
230 PRINT:INPUT"TOUR D'ARRIVEE :";A
240 B$=A$(D)
250 C$=A$(A)
260 IF MID$(B$,LEN(B$),1)>MID$(C$,LEN(C$
),1) THEN 200
270 A$(A)=C$+MID$(B$,LEN(B$),1)
280 A$(D)=LEFT$(B$,LEN(B$)-1)
290 PLAY"DO;RE;MI":GOTO 100
```



Losanges

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : interlude.
Difficulté du jeu : aucune.

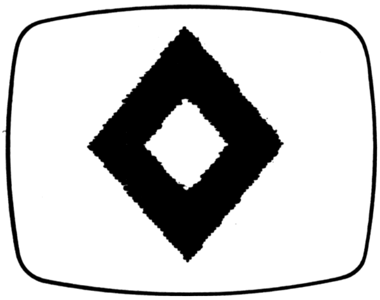
LE JEU :
Le programme va progressivement construire sur l'écran un losange, puis il va l'effacer progressivement, le reconstruire, et ainsi de suite.
Vous pourrez envisager de l'utiliser pour vous endormir, si les moutons ne vous suffisent plus.
Une version, très légèrement modifiée, vous est également proposée, qui vous permettra de faire apparaître des motifs aléatoires après la construction de chaque losange.

LE PROGRAMME :
Une première boucle K (5Ø-13Ø) définit les dimensions des tracés successifs.
La deuxième boucle I (6Ø-12Ø) trace des bandes de points, en utilisant des instructions PSET (lignes 8Ø à 11Ø).
Si la variable S (ligne 7Ø) est à 1, alors il y aura effacement (on utilise PSET, mais en se servant des couleurs du fond pour effacer).
On obtiendra dans le deuxième programme un effacement aléatoire en modifiant la ligne 7Ø.

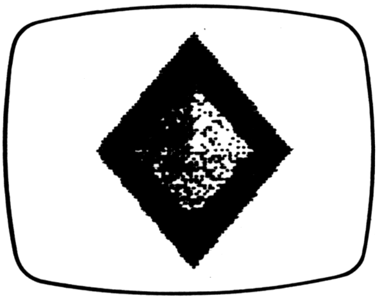
POUR JOUER :
On commence l'exécution du programme avec RUN. On entre si nécessaire un noyau pour le générateur (deuxième programme), et on regarde.
On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On pourra introduire le son, utiliser l'instruction LINE qui permettra un tracé beaucoup plus rapide, imaginer sur des principes analogues divers programmes (kaléidoscope, tapisserie...).

```
10 REM LOSANGES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:S=0
50 FOR K=0 TO 40
60 FOR I=0 TO K
70 IF S<>0 THEN ON INT(RND*5+1) GOTO 160
,170,180,190,200
80 PSET (150-K+I,100-I),1
90 PSET (150-K+I,100+I),1
100 PSET (150+K-I,100+I),1
110 PSET (150+K-I,100-I),1
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
160 PSET (150-K+I,100-I),-6
170 PSET (150-K+I,100+I),-5
180 PSET (150+K-I,100+I),-4
190 PSET (150+K-I,100-I),-3
200 GOTO 120
```



```
10 REM LOSANGES
40 CLS:S=0
50 FOR K=0 TO 40
60 FOR I=0 TO K
70 IF S<>0 THEN 160
80 PSET (150-K+I,100-I),1
90 PSET (150-K+I,100+I),1
100 PSET (150+K-I,100+I),1
110 PSET (150+K-I,100-I),1
120 NEXT I
130 NEXT K
140 S=NOT S
150 GOTO 50
160 PSET (150-K+I,100-I),-7
170 PSET (150-K+I,100+I),-7
180 PSET (150+K-I,100+I),-7
190 PSET (150+K-I,100-I),-7
200 GOTO 120
```



Mathieu

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin aléatoire.
Difficulté du jeu : aucune.

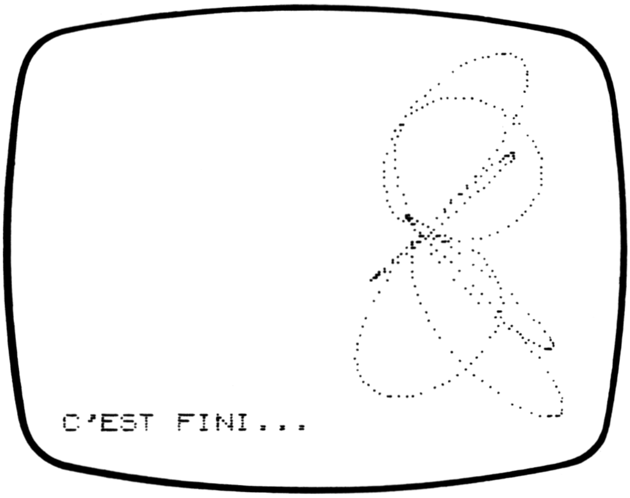
LE JEU :
On utilise la définition d'une courbe en coordonnées polaires et la fonction PSET pour obtenir sur l'écran des dessins aléatoires (de type fleur), plus ou moins élégants.
Les deux paramètres principaux A et B pourront être, si nécessaire, entrés à la main (instructions INPUT) pour faire une étude plus systématique des différentes courbes possibles.

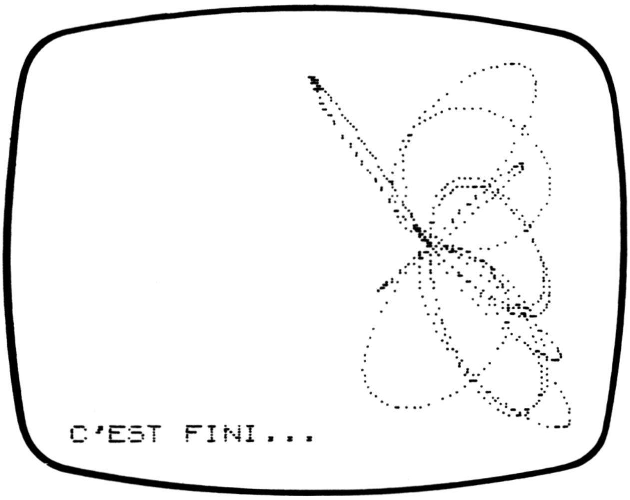
LE PROGRAMME :
Les 3 premiers paramètres Z, K et T, introduits en ligne 4Ø, définissent le rayon maximum et l'occupation de l'écran.
Les variables aléatoires A et B, choisies entre Ø et 3, de la ligne 6Ø, définissent l'allure de la courbe ; il pourra être intéressant de les introduire manuellement (par exemple : 6Ø INPUT A, B).
La boucle H (7Ø-12Ø) réalise le tracé, point par point. Le rayon R est calculé en ligne 8Ø, et le passage en coordonnées cartésiennes est effectué aux lignes 9Ø et 1ØØ.
L'affichage des points est réalisé en ligne 11Ø.
La ligne 13Ø imprime un message de fin.
La ligne 16Ø permet de refaire un nouveau dessin, alors que les lignes 17Ø et 18Ø assurent un nouveau tracé sans effacement du précédent.

POUR JOUER :
On lance le programme en tapant RUN.
Le dessin sera ensuite exécuté automatiquement.
Le message "c'est fini" signale la fin du tracé.
Si on veut un nouveau tracé, on tape Ø.
Si on veut un tracé sans effacement du dessin précédent, on tape une touche quelconque.
Pour arrêter, faire CNT et C.

EXTENSIONS POSSIBLES :
On pourra essayer d'utiliser d'autres fonctions pour réaliser des tracés encore plus extraordinaires (par exemple avec simulation de la troisième dimension).

```
10 REM MATHIEU
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 Z=7:K=9:T=3:CLS
50 C=INT(RND*8):F=7-C:SCREEN C,F
60 A=3*RND:B=3*RND
70 FOR H=0 TO 6.28 STEP 3.14/180
80 R=Z*SIN(H*T)
90 X=150+INT(K*R*COS(A*H))
100 Y=100+INT(K*R*SIN(B*H))
110 PSET (X,Y),C
120 NEXT H
130 LOCATE 0,20:PRINT"C'EST FINI..."
140 PLAY"DOMISO"
150 A$=INPUT$(1)
160 IF A$="0" THEN 40
170 C=INT(RND*8):IF C=F THEN 170
180 GOTO 60
```





Ordre

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

LE JEU :
Les 1Ø premières lettres de l'alphabet sont affichées dans le désordre ; il faut réussir à les remettre dans l'ordre, avec le moins de coups possible, par des permutations successives de deux des lettres ; mais ceci a aussi pour inconvénient de faire permuter autour d'elles les deux lettres qui entourent celles-ci, suivant le mécanisme expliqué ci-dessous :

B A D G H I J C F E
Ø 1 2 3 4 5 6 7 8 9 : permutation 83

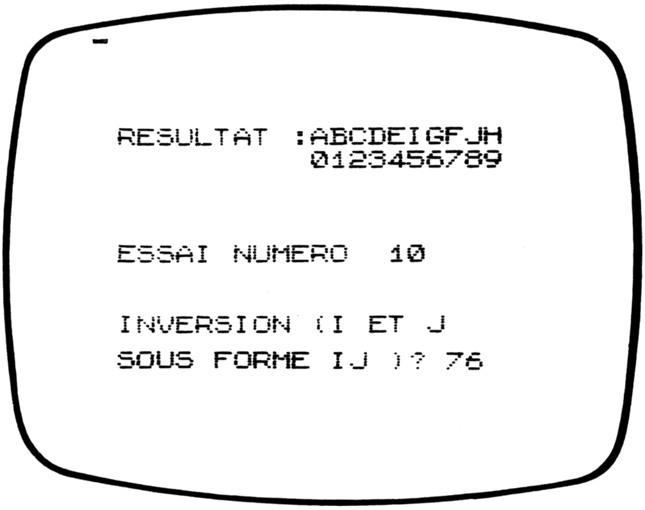
B A D F H I J E G C
Ø 1 2 3 4 5 6 7 8 9 : résultat

LE PROGRAMME :
Les boucles emboîtées I (6Ø-12Ø) et J (8Ø-1ØØ) définissent une variable alphanumérique A$, contenant les 1Ø premières lettres de l'alphabet dans le désordre.
La ligne 14Ø affiche A$, et la ligne 15Ø affiche les numéros de lettres de l'alphabet pour la permutation.
On entre la permutation en ligne 19Ø.
Les lignes 2ØØ et 21Ø transforment la variable alphanumérique entrée en une variable numérique.
La ligne 22Ø appelle une première fois le sous-programme d'inversion.
La ligne 25Ø appelle une deuxième fois le sous-programme d'inversion.
La ligne 26Ø comptabilise les essais.
La ligne 18Ø contrôle les résultats.

POUR JOUER :
Lancer le programme par RUN.
Le mot initial est affiché, chaque lettre étant repérée par un numéro de 1 à 9.
A la question INVERSION (I/J), on répond en entrant un nombre de deux chiffres xy ; x est le numéro de la première lettre, et y le numéro de la seconde. Les deux inversions sont alors successivement effectuées, et l'on recommence.
Le nombre de coups sera affiché en fin de partie.

EXTENSIONS POSSIBLES :
On peut imaginer un jeu contre l'ordinateur, qui évaluerait le score du joueur en fonction de ses propres performances ; on peut aussi modifier les règles qui régissent les permutations.

```
10 REM ORDRE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 A$=""
60 FOR I=1 TO 10
70 R$=CHR$(INT(RND*10+65))
80 FOR J=1 TO I-1
90 IF R$=MID$(A$,J,1) THEN 70
100 NEXT J
110 A$=A$+R$
120 NEXT I
130 S=1
140 LOCATE 0,4:PRINT"RESULTAT :";:COLOR
3,1:PRINT A$:COLOR 4,6
150 PRINT TAB(10);"0123456789"
160 LOCATE 0,9:PRINT"ESSAI NUMERO ";S
170 LOCATE 33,12:PRINT"        "
180 IF A$="ABCDEFGHIJ" THEN PRINT"TROUVE
 EN ";S;" COUPS":PLAY"DO;SO;MI":END
190 LOCATE 0,12:INPUT"INVERSION (I ET J
SOUS FORME IJ )";R$
200 I=ASC(LEFT$(R$,1))-47
210 J=ASC(RIGHT$(R$,1))-47
220 GOSUB 500
230 IF LEFT$(R$,1)="0" OR LEFT$(R$,1)="9
" THEN 250
240 I=I-1:J=I+2
250 GOSUB 500
260 S=S+1
270 GOTO 140
500 B$=""
510 FOR T=1 TO LEN(A$)
520 IF T=I THEN N$=MID$(A$,T,1)
530 IF T=J THEN M$=MID$(A$,T,1)
540 NEXT T
550 FOR T=1 TO LEN(A$)
560 IF T=I THEN B$=B$+M$:GOTO 590
570 IF T=J THEN B$=B$+N$:GOTO 590
580 B$=B$+MID$(A$,T,1)
590 NEXT T
600 A$=B$
610 RETURN
```


RESULTAT :ABCDEIGFJH
0123456789
ESSAI NUMERO 10
INVERSION (I ET J
SOUS FORME IJ)? 76

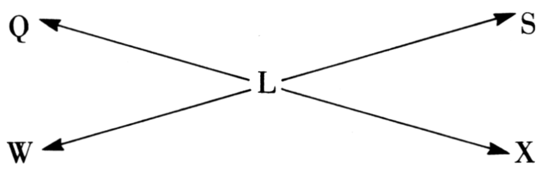
Le loup et les agneaux

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

LE JEU :
C'est un jeu assez connu qui peut être pratiqué indifféremment sur un damier 8 sur 8, ou 1Ø sur 1Ø. Il est présenté, ici, dans une version pour un damier de 8 cases ; l'un des joueurs choisit le loup qui se déplace suivant les diagonales, en restant sur les cases noires, d'une case à chaque coup, et qui peut avancer ou reculer. Les agneaux se déplacent également sur les cases noires, en diagonale, mais n'ont pas le droit de reculer. Les cases déjà occupées sont interdites (pas de prises), et il n'y a pas de saut.
Le loup gagne s'il réussit à arriver sur la ligne du bas avant que tous les agneaux n'occupent celle du haut ; les agneaux gagnent dans ce dernier cas, ou s'ils arrivent à bloquer le loup.

LE PROGRAMME :
Une variable A$, en ligne 3Ø contient les positions de départ.
La boucle I (8Ø-13Ø) dessine le damier, et les lignes 15Ø à 19Ø placent le loup et les agneaux.
La ligne 2ØØ gère le changement de joueur.
La ligne 22Ø affiche JEU DU LOUP si c'est le joueur qui manipule le loup qui a le trait, et la ligne 23Ø JEU DES AGNEAUX dans le cas contraire.
Les lignes 25Ø à 27Ø analysent la réponse, en fonction du joueur concerné.

POUR JOUER :
RUN déclenche l'arrêt sur l'attente d'une entrée : Ø et ENTREE si l'on veut que les agneaux commencent, et 1 pour le loup. Après quoi, le damier s'affiche et la phrase JEU DU LOUP (Q, S, W, X) pour le loup et JEU DES AGNEAUX (ND et NG) pour les agneaux.
Les agneaux répondent par un numéro, suivi de D ou G (pour droite ou gauche).
Ainsi 2D provoque le déplacement de l'agneau numéro 2 qui monte en diagonale d'une case sur la droite. Le déplacement effectué est affiché, puis on obtient une question identique pour le loup qui répondra en tapant Q, S, W ou X, suivant la direction choisie :



On devra veiller à ne pas sortir de l'échiquier et à ne pas venir sur l'adversaire, aucun contrôle n'ayant été prévu.

EXTENSIONS POSSIBLES :

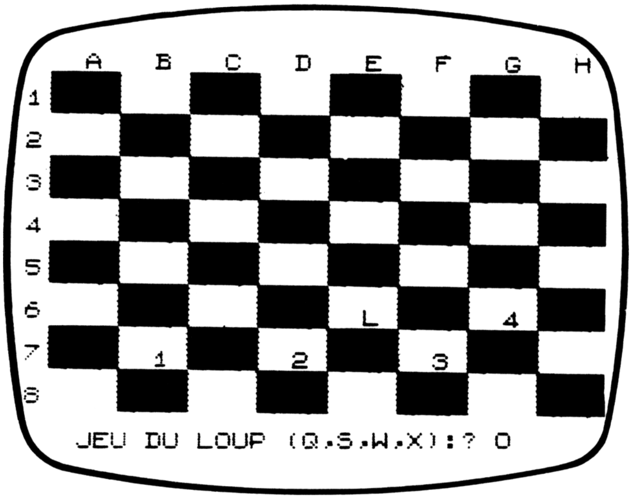
On pourra prévoir un repérage des cases, un graphisme plus soigné, des contrôles de non dépassement des limites du terrain, de non venue sur une case déjà occupée...

On pourra aussi envisager de jouer sur un damier de 1Ø cases sur 1Ø cases, ce qui augmente légèrement la difficulté du jeu pour les agneaux. Mais l'extension la plus intéressante à réaliser sera certainement la réalisation d'un programme permettant au MO6 de se comporter en adversaire honorable, soit avec le loup, soit avec les agneaux (et, dans ce cas, il sera possible de définir une stratégie gagnante pour le MO6).

```
10 REM LE LOUP ET LES AGNEAUX
20 INPUT"AGNEAUX (0) OU LOUP (1)";S
30 A$="8183858714"
40 CLS
50 PRINT"     A   B   C   D   E   F   G
  H"
60 FOR I=1 TO 8:LOCATE 0,2*I:PRINT I:NEX
T I
70 ATTRB 1,1:COLOR 3,1
80 FOR I=1 TO 8
90 FOR J=1 TO 8
100 LOCATE 4*J-1,2*I
110 IF (I+J)/2=INT((I+J)/2) THEN PRINT C
HR$(127)+CHR$(127) ELSE PRINT"  "
120 NEXT J
130 NEXT I
140 ATTRB 0,0:COLOR 4,6
150 LOCATE 4*VAL(MID$(A$,10,1))+1,2*VAL(
MID$(A$,9,1)):PRINT "L"
160 LOCATE 4*VAL(MID$(A$,2,1))+1,2*VAL(M
ID$(A$,1,1)):PRINT"1"
170 LOCATE 4*VAL(MID$(A$,4,1))+1,2*VAL(M
ID$(A$,3,1)):PRINT"2"
180 LOCATE 4*VAL(MID$(A$,6,1))+1,2*VAL(M
ID$(A$,5,1)):PRINT"3"
190 LOCATE 4*VAL(MID$(A$,8,1))+1,2*VAL(M
ID$(A$,7,1)):PRINT"4"
200 S=ABS(S-1)
210 LOCATE 0,20:PRINT"
                   "
220 LOCATE 0,20:IF S=0 THEN INPUT"JEU DU
 LOUP (Q,S,W,X):";R$
230 IF S=1 THEN INPUT"JEU DES AGNEAUX (N
D OU NG):";R$:GOTO 280
240 C=10
250 A=VAL(RIGHT$(A$,2))+(C+1 AND R$="X")
+(C-1 AND R$="W")
```

→

```
260 A=A-(C+1 AND R$="Q")-(C-1 AND R$="S"
)
270 GOTO 300
280 C=2*VAL(MID$(R$,1,1))
290 A=VAL(MID$(A$,C-1,2))-(9 AND RIGHT$(
R$,1)="D")-(11 AND RIGHT$(R$,1)="G")
300 A$=LEFT$(A$,C-2)+RIGHT$(STR$(A),2)+R
IGHT$(A$,LEN(A$)-C)
310 GOTO 70
```



Niveau 4

Nom du programme	*Thèmes et particularités*
Tirage de dés	Suffisamment souple pour être utilisé partout.
Jeu de loto	Gagnez sans problème, au prochain tirage.
La bombe	Saurez-vous la découvrir à temps ? (1 joueur).
Ardoise magique	Mettez au point vos créations graphiques, sans papier ni crayon.
Affichage publicitaire	Si vous ne savez pas quoi mettre dans votre vitrine.
Evasion	Il vous faudra beaucoup de chance pour réussir.
Chapeaux	Soyez très observateur, et vous réussirez (2 joueurs).
Combat aérien	Pas si facile de descendre un appareil qui bouge tout le temps.
Dessin de dés	Améliorez vos jeux de dés en incorporant ce programme.
Le jeu du missile	Une mission bien périlleuse (un seul joueur).
Le singe	Un jeu de lettres pour 2 joueurs, où il faut de bons réflexes.
Le jeu du 15	Un jeu de réflexion utilisant des chiffres (2 joueurs).
Adresse	Jeu de réflexe pour 1 joueur, très visuel.
Sous-marin	Un classique des jeux vidéo (1 joueur).
Zombies	Un peu farfelu, mais très visuel (1 joueur).
Diago	Etes-vous vraiment sûr de vos réflexes ?
Décomaque	Pour vous éviter de sécher trop longtemps sur "Nicomaque".
Chasse au sous-marin	Il vous faudra pas mal de réflexion, et beaucoup de chance.
Clavier	Apprenez à taper à la machine en vous amusant.
Le lézard d'Alice	Beaucoup de hasard, mais très spectaculaire.
Calculatrice	Un programme de simulation très réaliste.

Les techniques d'animation élémentaire

SAISIE AU VOL D'UN CARACTERE FRAPPE AU CLAVIER : INKEY$

Le principe d'un jeu graphique, en animation, est de laisser le jeu se dérouler tout seul, d'une certaine manière, pendant un certain temps, tant qu'aucune intervention n'a été faite par le joueur.

Il faut donc disposer de la possibilité d'observer s'il y a eu ou non intervention du joueur, sans pour cela interrompre le déroulement du programme.

Dans le Basic du MO6, cette propriété est offerte, par la **fonction INKEY$** qui effectue une scrutation des touches du clavier et enregistre le nom de la dernière touche frappée.

On pourra utiliser INKEY$ de deux façons :

- soit dans une boucle (réalisation d'un mouvement). On saisit INKEY$ et on le teste pour une éventuelle sortie de la boucle.
- soit dans une attente : tant que INKEY$ est vide (on n'a rien frappé) on attend, et l'on sort dès qu'une touche a été frappée.

Il sera quelquefois nécessaire de taper plusieurs fois la touche pour qu'elle soit prise en compte.

DEPLACEMENTS D'OBJETS DE BAS EN HAUT SUR L'ECRAN : SCROLL

Il est difficile, si l'on travaille en Basic, de gérer le déplacement simultané de plusieurs objets sur l'écran (sinon on obtient vite une animation saccadée, à cause d'un temps de déplacement prohibitif).

Le MO6 réalise automatiquement un "**Scrolling**" (c'est-à-dire un déplacement de l'ensemble de l'image) vers le haut, lorsque l'on écrit sur la dernière ligne en bas de l'écran.

Cette propriété a été mise à profit dans plusieurs jeux : cela permet en effet, de déplacer d'une ligne vers le haut les divers objets (textes ou graphiques) présents sur l'écran à un moment donné. La ligne du haut est bien évidemment perdue.

Si un objet doit rester immobile sur l'écran, il sera nécessaire de l'effacer dans sa nouvelle position, et de le replacer à son emplacement initial.

LE GRAPHISME DE HAUT NIVEAU : L'INSTRUCTION DEFGR$ (I)

Le MO6 dispose de la possibilité laissée à l'utilisateur de créer lui-même ses propres caractères, dans une matrice 8 × 8. Pour cela, on réserve de la place par une instruction CLEAR, ,n (n étant le nombre de graphismes à créer : on peut en faire jusqu'à 128). Puis chaque dessin est défini par 8 nombres, de Ø à 255, dans une instruction DEFGR$ (I).

Exemple : DEFGR$ (Ø) = 13,25,64,12,Ø,29,62,13.

Les dessins peuvent être ensuite utilisés dans des PRINT, soit par PRINT GR$ (I), soit par PRINT CHR$ (128 + I).

Tirage de dés

REFERENCES :
Difficulté de programmation : aucune.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : aucune.

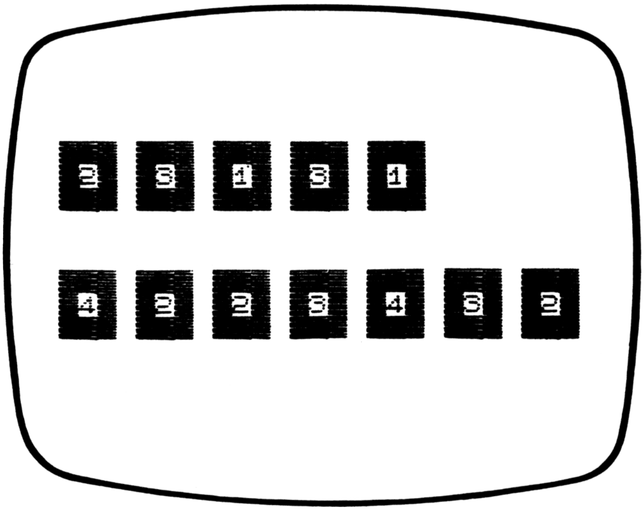
LE JEU :
Le programme proposé ne réalise pas un jeu en lui-même, mais pourra être utilisé en sous-programme pour de nombreux jeux.
Il offre simplement la possibilité de lancer de 1 à 9 dés, par simple pression d'une touche. Les dés sont ensuite affichés avec un graphisme très simplifié.

LE PROGRAMME :
La ligne 6Ø saisit le caractère frappé au clavier et le transforme en nombre (si ce n'est pas possible, le programme refuse l'entrée).
Une variable D non dimensionnée reçoit la valeur indiquée par chaque dé.
Une boucle J (11Ø-13Ø) est utilisée pour dessiner un fond, et une autre boucle I (1ØØ-16Ø) sert au tirage (tirage en ligne 14Ø et impression en ligne 15Ø).
La ligne 17Ø offre la possibilité de refaire un tirage ; elle n'est pas indispensable.

POUR JOUER :
On lance le programme par l'ordre RUN et on obtient l'affichage NOMBRE DE DES A LANCER auquel on répond en frappant un chiffre.
Le résultat est alors automatiquement affiché ; il suffit de presser une touche quelconque pour recommencer le programme.
Pour arrêter, il est nécessaire de taper CNT et C.

EXTENSIONS POSSIBLES :
On pourra, bien sûr, utiliser le canevas fourni par ce programme pour la réalisation de tout jeu se servant de dés (421, YAM'S, CRAPS, etc.).

```
10 REM TIRAGE DE DES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 LOCATE 0,0:PRINT"NOMBRE DE DES A LANC
ER :"
60 A$=INKEY$:IF A$="" THEN 60
70 IF A$<"1" OR A$>"9" THEN 40
80 N=VAL(A$)
90 CLS
100 FOR I=1 TO N
110 FOR J=1 TO 3
120 LOCATE 4*I-4,8+J:COLOR 3,1:PRINT "
 "
130 NEXT J
140 D(I)=INT(RND*6+1)
150 LOCATE 4*I-3,10:COLOR 2,0:PRINT RIGH
T$(STR$(D(I)),1)
160 NEXT I
170 COLOR 4,6:PLAY"DO;RE;MI":GOTO 50
```



Jeu de loto

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : hasard (plusieurs joueurs).
Difficulté du jeu : facile.

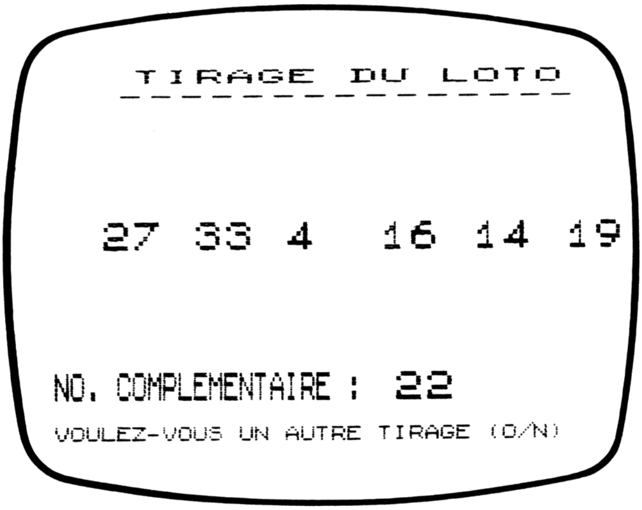
LE JEU :
Il s'agit de réaliser un tirage complet de loto, c'est-à-dire d'obtenir 6 numéros, compris entre 1 et 49, plus un numéro complémentaire, tous différents, qui pourraient être, par exemple, utilisés pour remplir votre prochaine grille de loto.

LE PROGRAMME :
On utilisera un tableau N(7), dimensionné en ligne 4Ø, pour enregistrer les résultats.
Le tirage des nombres est effectué dans une boucle I (6Ø-12Ø), une deuxième boucle J (9Ø-11Ø) étant utilisée pour contrôler qu'il n'y a pas deux tirages identiques.
Le tirage (ou le retirage) est effectué en ligne 7Ø.
La boucle I (16Ø-19Ø) affiche le résultat, et la ligne 2ØØ affiche le numéro complémentaire.
Les lignes 22Ø et 23Ø s'occupent du cas où l'on voudrait refaire une autre partie.

POUR JOUER :
Il suffit de taper RUN pour lancer le programme et obtenir l'affichage d'un premier loto.
La question VOULEZ-VOUS UN AUTRE TIRAGE (O/N) s'affiche en bas de l'écran ; on répondra O si l'on désire continuer.

EXTENSIONS POSSIBLES :
On peut imaginer un programme qui pourrait tenir compte, soit de ses propres résultats, soit de résultats entrés par un joueur, pour améliorer le tirage, en tenant compte des probabilités de sortir un numéro donné.
On peut aussi envisager un programme qui réaliserait des cartons pour le jeu de loto plus usuel, tel qu'il est encore couramment pratiqué dans le Midi de la France.

```
10 REM JEU DE LOTO
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 DIM N(7)
50 CLS
60 FOR I=1 TO 7
70 N(I)=INT(RND*49+1)
80 IF I=1 THEN 120
90 FOR J=1 TO I-1
100 IF N(I)=N(J) THEN 70
110 NEXT J
120 NEXT I
130 LOCATE 5,3:COLOR 5,6:ATTRB 1,0
140 PRINT"TIRAGE DU LOTO"
150 PRINT"  ---------------"
160 FOR I=1 TO 6
170 LOCATE 6*I-5,12:COLOR 1,7:ATTRB 1,1
180 PRINT N(I)
190 NEXT I
200 LOCATE 0,20:COLOR 0,6:ATTRB 0,1:PRIN
T"NO. COMPLEMENTAIRE :";:ATTRB 1,1:PRINT
 N(7)
210 LOCATE 0,22:COLOR 4,6:ATTRB 0,0:PLAY
"DO;RE;MI"
220 PRINT"VOULEZ-VOUS UN AUTRE TIRAGE (O
/N)";:A$=INPUT$(1)
230 IF A$="O" THEN 50
```



La bombe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

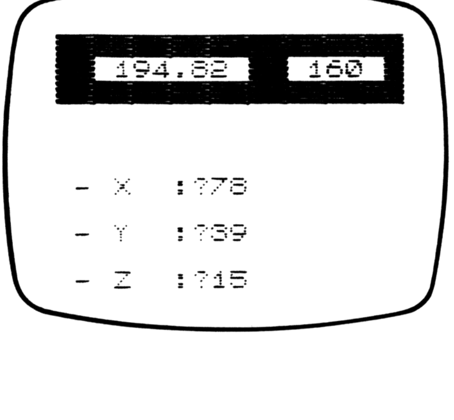
LE JEU :
Vous êtes censé trouver, à l'aide d'un détecteur spécialisé, un colis radioactif piégé disposé par un voyageur mal intentionné, dans une chambre d'hôtel. Seul problème, mais de taille : l'hôtel comporte un million de chambres (1ØØ*1ØØ*1ØØ).
Votre détecteur affiche, en permanence, un signal qui vous renseigne sur la proximité de la bombe, et le temps qui vous reste est affiché en permanence, décompté de 1Ø en 1Ø.
Pourrez-vous découvrir la bombe avant l'explosion ?

LE PROGRAMME :
Une variable numérique R, non dimensionnée, contient les trois coordonnées définissant la position du joueur.
La position de la chambre contenant le colis piégé est définie par un tirage aléatoire de 3 variables A, B, C, en ligne 5Ø.
La boucle I (7Ø-11Ø) dessine le détecteur.
Le signal de proximité est calculé en ligne 13Ø et la ligne 14Ø permet l'affichage des coordonnées de la chambre à découvrir, dans le cas où l'on a perdu.
La ligne 15Ø permet l'affichage, à la fois du signal et du temps restant à courir.
La ligne 16Ø réalise le décompte du temps.
La ligne 17Ø contrôle si l'on a trouvé ou non.
La boucle I (21Ø-25Ø) permet d'entrer les nouvelles coordonnées, en utilisant le sous-programme placé aux lignes 3ØØ à 34Ø.

POUR JOUER :
Après le lancement du programme par RUN, le signal et le temps (en principe 2ØØ) sont affichés.
Le joueur est alors en position x=Ø, y=Ø, z=Ø, et on devra introduire successivement les trois coordonnées x, y, z de la chambre à explorer.
On fera, par exemple, 13 puis ENTREE, 45 puis ENTREE, etc.
Le nouveau signal est alors affiché, et le temps est décrémenté de 1Ø.
On continuera ainsi, soit jusqu'au bon résultat (arrêt du jeu et fin de programme), soit jusqu'à l'apparition de 3 nombres (x,y,z) sans sortie du programme, ce qui signifie alors que l'on a perdu.

EXTENSIONS POSSIBLES :
Il sera sans doute indispensable d'améliorer le graphisme, d'ajouter une explosion (sonore et visuelle) pour symboliser l'éclatement de la bombe, d'afficher symboliquement la position dans l'hôtel, etc.

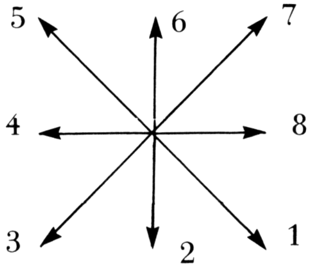
```
10 REM LA BOMBE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 L=100:T=200
50 A=INT(RND*L):B=INT(RND*L):C=INT(RND*L
)
60 CLS:SCREEN 1,0,4
70 FOR I=0 TO 17
80 LOCATE I,4:COLOR 5,0:PRINT CHR$(127)
90 LOCATE I,5:PRINT CHR$(127)
100 LOCATE I,6:PRINT CHR$(127)
110 NEXT I
120 S=ABS((A/L+B+C*L)-(R(1)/L+R(2)+R(3)*
L))
130 S$=LEFT$(STR$(S),7):S=VAL(S$)
140 IF T=0 THEN PRINT"PERDU . LA SOLUTIO
N :";A;"  ";B;"  ";C:PLAY"SI;SO;DO"
150 COLOR 0,2:LOCATE 2,5:PRINT S:LOCATE
12,5:PRINT T
160 PLAY"DO;RE;MI":T=T-INT(SQR(L))
170 IF S=0 THEN LOCATE 0,21:COLOR 4,6:PR
INT"BRAVO":PLAY"DO;SO;MI":END
180 LOCATE 0,10:PRINT" - X  :"
190 LOCATE 0,12:PRINT" - Y  :"
200 LOCATE 0,14:PRINT" - Z  :"
210 FOR I=1 TO 3
220 R$=""
230 GOSUB 300
240 R(I)=VAL(R$)
250 NEXT I
260 GOTO 70
300 COLOR 7,5:LOCATE 7,2*I+8:PRINT "?";R
$;:COLOR 7,0:PRINT"      ":BEEP
310 A$=INKEY$:IF A$="" THEN 310
320 IF ASC(A$)=13 THEN RETURN
330 R$=R$+A$
340 GOTO 300
```



Ardoise magique

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : dessin.
Difficulté du jeu : facile.

LE JEU :
Comme le titre peut sembler l'indiquer, il s'agit, tout simplement d'utiliser l'écran du téléviseur comme une ardoise, en manipulant un point sur l'écran suivant 8 directions, à l'aide des touches de 1 à 8, suivant la convention :



On dispose aussi de la possibilité de déplacer le point sur l'écran sans laisser de trace en appuyant sur la touche Ø (ce qui permet l'effacement partiel) et de laisser de nouveau une trace en réappuyant sur Ø.
On peut choisir la couleur du point en appuyant sur une touche de A à H ; et il est possible d'effacer tout l'écran en appuyant sur P.

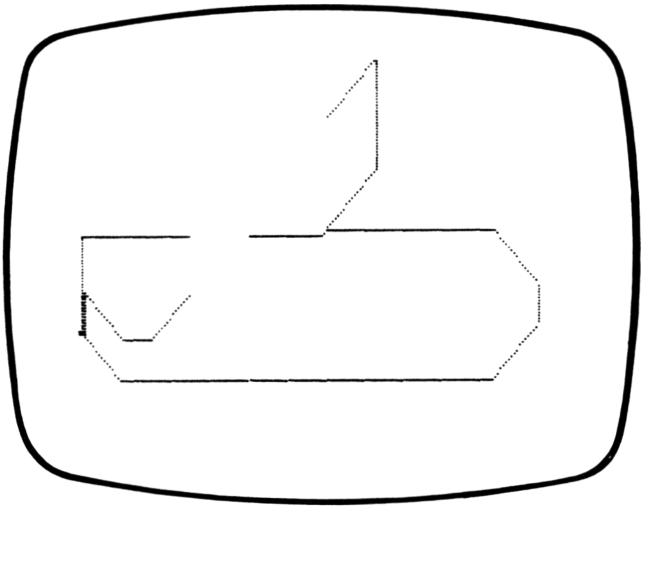
LE PROGRAMME :
La ligne 3Ø contient le choix des possibilités de couleur.
La ligne 5Ø imprime le point élémentaire.
Les lignes 6Ø à 9Ø contrôlent les entrées.
La ligne 13Ø appelle l'un des quatre sous-programmes effectuant le déplacement.
La ligne 14Ø vérifie que l'on ne sort pas des limites de l'écran.
La ligne 16Ø effectue l'effacement s'il y a lieu.
Le sous-programme 4ØØ-41Ø choisit la couleur ; 3ØØ-31Ø réalise ou non l'effacement.

POUR JOUER :
On lance le programme par RUN ; un point apparaît vers le milieu de l'écran. On devra taper Ø pour le déplacer sans afficher.
Si on frappe une lettre de A à H, la couleur du point change.
La touche P efface tout, et l'on recommence.
On peut sortir à tout moment, en tapant CNT et C.

EXTENSIONS POSSIBLES :

On pourra, bien sûr, envisager d'écrire un programme identique en utilisant des tracés de segments, prévoir un sous-programme qui permette la sauvegarde du dessin réalisé pour une utilisation ultérieure, etc.

```
10 REM ARDOISE MAGIQUE ( VERSION 1 )
20 COLOR 4,6:CLS
30 C$="01234567"
40 X=160:Y=100:C=4
50 PSET (X,Y),C
60 A$=INKEY$:IF A$="" THEN 60
70 IF A$>="A" AND A$<="H" THEN GOSUB 400
80 IF A$="0" THEN GOSUB 300
90 IF A$="P" THEN SCREEN 4,6:GOTO 20
100 F=-2*(VAL(A$)<5)-1
110 A=X:B=Y
120 U=VAL(A$)+2*(F-1)
130 ON U GOSUB 1000,2000,3000,4000
140 D=-(X>0 AND X<319 AND Y>0 AND Y<199)
150 IF D=0 THEN GOSUB 5000
160 IF S<>0 THEN PSET (A,B),6
170 GOTO 50
300 S=ABS(S-1)
310 RETURN
400 C=VAL(MID$(C$,ASC(A$)-64,1))
410 RETURN
1000 Y=Y+F
2000 X=X+F
2010 RETURN
3000 GOSUB 2000
4000 Y=Y-F
4010 RETURN
5000 X=A:Y=B
5010 RETURN
```



Affichage publicitaire

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : affichage publicitaire.
Difficulté du jeu : aucune.

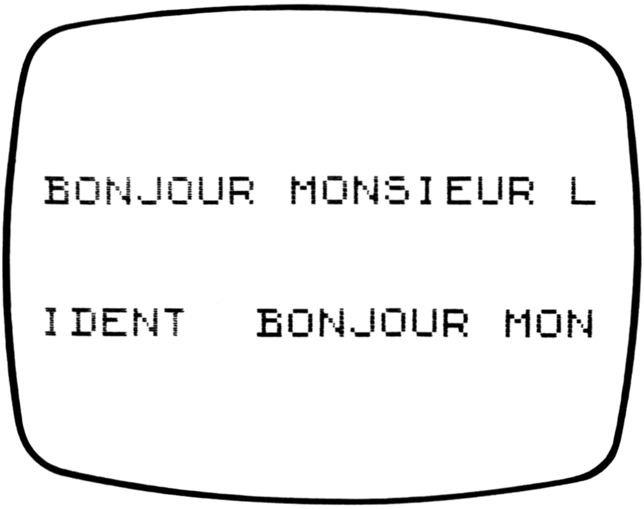
LE JEU :
Un texte quelconque devra être introduit en début de programme. Il pourra être relativement long (4 ou 5 lignes) ; on choisira ensuite le sens et la vitesse de défilement, à la suite de quoi le texte va apparaître (une seule ligne à la fois) et commencera à défiler, à la vitesse choisie, et dans le sens choisi.

LE PROGRAMME :
Le texte sera introduit lettre par lettre aux lignes 6Ø à 9Ø.
L'affichage sera tronqué à 18 caractères au plus, grâce à la ligne 17Ø.
Le défilement vers la gauche est préparé à la ligne 18Ø, et le défilement vers la droite à la ligne 19Ø.
Le texte sera affiché en ligne 17Ø, et la durée de l'affichage (vitesse de défilement) est choisie par la ligne 2ØØ.
La ligne 22Ø permettra de changer, si besoin est, les conditions du défilement.

POUR JOUER :
On lancera le programme par RUN, et à l'affichage de la question ENTREZ VOTRE TEXTE, on introduira le texte à faire défiler, suivi de ENTREE.
On obtient ensuite l'affichage DEFILEMENT A DROITE (D) OU A GAUCHE (G) auquel on répondra en tapant, soit D, soit G, suivi de ENTREE.
Puis à la question VITESSE, on répondra en introduisant un petit nombre pour une vitesse lente, et un grand pour une vitesse rapide (par exemple, de Ø à 4Ø ou plus).
Le texte commencera à défiler dès que l'on aura frappé ENTREE.
Si l'on désire changer les conditions du défilement, il suffit d'appuyer sur la touche Ø pour retrouver les questions SENS et VITESSE. On tapera CNT et C pour interrompre le programme.

EXTENSIONS POSSIBLES :
On pourra introduire de la couleur, réaliser un affichage ondulant, faire remonter les lignes sur l'écran, etc.

```
10 REM AFFICHAGE PUBLICITAIRE
20 CLS
30 B$=""
40 PRINT"ENTREZ VOTRE TEXTE"
50 PRINT
60 A$=INKEY$:IF A$="" THEN 60
70 IF ASC(A$)=13 THEN 100
80 B$=B$+A$
90 PRINT A$;:GOTO 60
100 ATTRB 0,0:PRINT:PRINT"DEFILEMENT A D
ROITE (D) OU A GAUCHE (G)":INPUT S$
110 PRINT
120 INPUT"VITESSE (DE 0 A 50) :";V
130 V=51-V
140 CLS:SCREEN 1,0,5
150 T=LEN(B$)
160 LOCATE 0,12:ATTRB 1,1
170 PRINT" "+MID$(B$,1,18)+" "
180 IF LEFT$(S$,1)="G" THEN B$=RIGHT$(B$
,T-1)+LEFT$(B$,1)
190 IF LEFT$(S$,1)="D" THEN B$=RIGHT$(B$
,1)+LEFT$(B$,T-1)
200 FOR J=0 TO 10*V
210 C$=INKEY$
220 IF C$="0" THEN 100
230 NEXT J
240 GOTO 160
```



Evasion

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : facile.

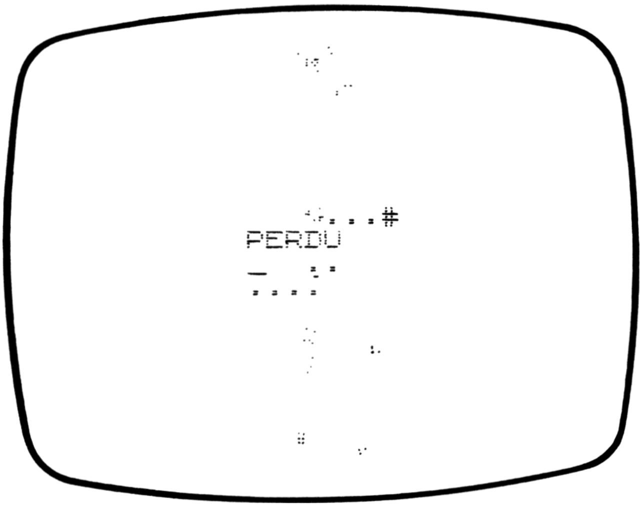
LE JEU :
Votre seul problème : réussir à traverser l'écran, sans que le MO6 vous ait auparavant touché par l'un des nombreux obus explosifs qu'il va tirer sur vous.
Vous pouvez vous déplacer dans 4 directions (à l'aide des 4 flèches), et vous devez réussir à sortir du terrain par le bord droit, sans, bien sûr, avoir été touché une seule fois.

LE PROGRAMME :
La ligne 5Ø définit la position de départ du joueur.
La ligne 6Ø choisit la coordonnée horizontale de la case où MO6 va tirer (liée à votre position), et la ligne 7Ø choisit la coordonnée verticale (arbitraire) de cette case.
Le déplacement du joueur est contrôlé par les lignes 14Ø et 15Ø ; sa trajectoire est tracée par la ligne 12Ø, et sa position actuelle (sur l'écran, un #) est imprimée par la ligne 16Ø.
La ligne 17Ø regarde s'il est touché ou non.
La ligne 18Ø effectue le contrôle de fin de partie.

POUR JOUER :
On commence l'exécution du programme par RUN.
Un premier tir a lieu, et ce n'est qu'après que le joueur apparaît, représenté par un #.
Les 4 touches de direction (←, ↓ , ↑ ,→) servent à déplacer le joueur.
S'il est touché, on obtient, sur l'écran, le message PERDU et la partie s'arrête.
S'il réussit à sortir par la droite de l'écran sans être touché, on obtient le message BRAVO, et la partie est également terminée.
On peut terminer à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
La couleur et un graphisme plus soigné amélioreront considérablement l'attrait du jeu. On pourra aussi rendre le tir des bombes plus aléatoire, introduire des obstacles à contourner, ajouter des contrôles, faire participer plusieurs joueurs, etc.

```
10 REM EVASION
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 X=0:Y=12
60 H=8*(X-1)+16*RND
70 K=190*RND+5
80 IF X<3 THEN 120
90 FOR J=1 TO 7*RND+3
100 PSET (H+6*RND-2,K+6*RND-2),1
110 NEXT J
120 COLOR 4,6:BEEP:BEEP:LOCATE X,Y:PRINT
 "."
130 A$=INKEY$
140 X=X-(A$=CHR$(9))+(A$=CHR$(8))
150 Y=Y-(A$=CHR$(10))+(A$=CHR$(11))
160 LOCATE X,Y:PRINT "#"
170 IF X>3 AND (ABS(H-8*X)+ABS(K-8*(25-Y
))<=10) THEN PRINT"PERDU":PLAY"SI;SO;DO"
:END
180 IF X<38 THEN 60
190 PRINT"BRAVO":PLAY"DO;MI;SO"
```



Chapeaux

REFERENCES :
Difficulté de programmation : assez facile.
Catégorie de jeu : jeu de réflexe, pour deux joueurs.
Difficulté du jeu : facile.

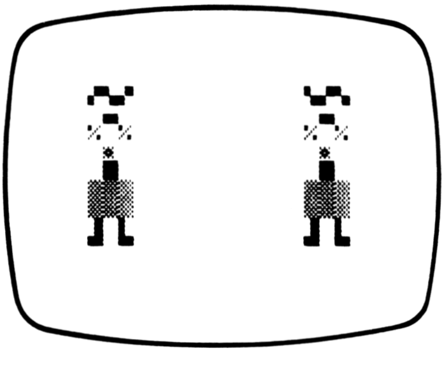
LE JEU :
L'idée en est très simple : deux personnages se déplacent l'un vers l'autre sur l'écran et se croisent. Au moment où ils se croisent, il est possible qu'ils échangent leurs chapeaux. Il s'agit d'être le premier à dire si oui ou non il y a eu échange de chapeaux lors du croisement.

LE PROGRAMME :
La ligne 4Ø contient le dessin des deux chapeaux (dans deux variables alphanumériques B$ et C$).
La ligne 5Ø contient, dans une autre variable A$, les symboles qui permettront de dessiner les personnages.
La ligne 7Ø permet, s'il y a lieu, l'échange des chapeaux.
La variable T de la ligne 8Ø sert à repérer s'il y a eu échange ou non.
Le dessin des deux personnages est réalisé par une boucle I (12Ø-15Ø), et leur déplacement est réalisé par la ligne 16Ø.
Le test, permettant ou non le changement des couvre-chefs, est effectué en ligne 18Ø.
Les lignes 2ØØ à 26Ø déterminent si le premier joueur ayant appuyé sur une touche a gagné ou perdu.
Sur TO8 et TO9+, il sera préférable de modifier les touches utilisées.

POUR JOUER :
Lancer le programme par RUN.
Les personnages commencent immédiatement à se déplacer l'un vers l'autre.
Le joueur de gauche dispose des touches 1 et 2 et celui de droite, des touches 9 et Ø.
S'il y a eu changement de chapeau, les touches à appuyer seront 1 (à gauche) et Ø (à droite).
S'il n'y a pas eu de changement, il faudra, pour gagner, être le premier à appuyer sur 2 (à gauche) et sur 9 (à droite).
L'affichage indiquera alors BRAVO GAUCHE ou BRAVO DROITE, ou, encore, PERDU GAUCHE ou PERDU DROITE.

EXTENSIONS POSSIBLES :
Il sera possible de prévoir un affichage couleur, avec une animation des personnages.
On pourra aussi compliquer le jeu à plaisir avec plusieurs objets échangés de façon aléatoire (il faudrait alors en trouver la liste complète).

```
10 REM CHAPEAUX
15 INPUT"DONNEZ UN ENTIER :";F
20 FOR I=1 TO F:Y=RND:NEXT I
25 CLEAR,,7:DEFGR$(0)=15,15,15,15,240,24
0,240,240:DEFGR$(1)=240,240,240,240,15,1
5,15,15:DEFGR$(2)=240,240,240,240,240,24
0,240,240
30 DEFGR$(3)=15,15,15,15,15,15,15,15:DEF
GR$(4)=255,255,255,255,0,0,0,0:DEFGR$(5)
=170,85,170,85,170,85,170,85:DEFGR$(6)=0
,0,0,0,255,255,255,255
40 B$=GR$(0)+GR$(1)+GR$(0):C$=GR$(1)+GR$
(0)+GR$(1)
50 A$=" "+GR$(6)+" % % *  "+CHR$(127)+"
"+GR$(5)+GR$(5)+GR$(5)+GR$(5)+GR$(5)+GR$
(5)+GR$(3)+" "+GR$(2)+GR$(4)+" "+GR$(4)
60 X=0:Y=30:T=1
70 D$=C$:C$=B$:B$=D$
80 T=ABS(T-1)
90 CLS:SCREEN 4,6
100 LOCATE X,0:PRINT C$
110 LOCATE Y,0:PRINT B$
120 FOR I=1 TO 8
130 COLOR 1,6:LOCATE X,I,0:PRINT MID$(A$
,3*I-2,3)
140 COLOR 5,6:LOCATE Y,I,0:PRINT MID$(A$
,3*I-2,3)
150 NEXT I
160 X=X+1:Y=Y-1
170 FOR I=1 TO 200:NEXT I
180 IF (RND>.5) AND X-2=Y THEN 70
190 R$=INKEY$:IF R$="" THEN 90
200 IF X<=Y THEN 240
210 COLOR 4,6
220 IF (R$="1" AND T=1) OR (R$="2" AND T
=0) THEN PRINT "BRAVO GAUCHE":PLAY"DO;MI
;SO":END
230 IF (R$="0" AND T=1) OR (R$="9" AND T
=0) THEN PRINT "BRAVO DROIT":PLAY"DO;RE;
MI":END
240 IF R$="2" OR R$="1" THEN PRINT "PERD
U GAUCHE"
250 IF R$="9" OR R$="0" THEN PRINT "PERD
U DROIT"
260 PLAY"DO;DO;DO"
```



Combat aérien

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 joueur).
Difficulté du jeu : difficile.

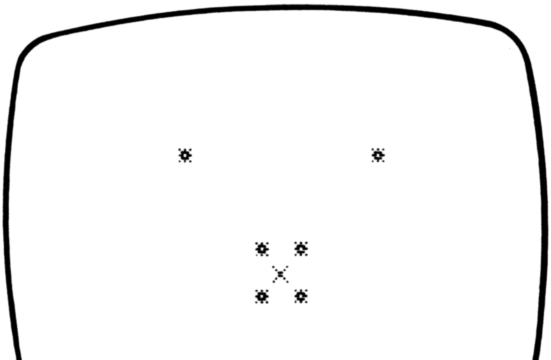
LE JEU :
Au centre de l'écran, se trouve le viseur de votre appareil, et l'avion ennemi se déplace devant vous, d'une façon qui pourra vous sembler au début incohérente.
Vous pouvez vous déplacer dans toutes les directions, grâce aux flèches de mouvement, et vous allez essayer d'amener l'appareil ennemi au centre de votre viseur.
Vous tirez sur lui en frappant Ø, et vous voyez vos projectiles qui convergent sur lui. Mais il continue à se déplacer, même pendant votre tir, et il n'est pas du tout évident que vous réussirez à l'avoir.
La partie continuera à se dérouler jusqu'à ce que vous l'ayez touché (affichage de GAGNE) ou qu'il soit sorti de votre champ de vision.

LE PROGRAMME :
La ligne 7Ø dessine l'appareil ennemi dont le déplacement aléatoire sera réalisé par les lignes 9Ø et 1ØØ.
Les lignes 12Ø et 13Ø vous permettent le pilotage de votre appareil.
La ligne 17Ø dessine le viseur de votre appareil.
Le sous-programme, gérant le tir, va de la ligne 2ØØ à la ligne 25Ø et le test de fin de partie est effectué en ligne 24Ø.
La ligne 18Ø regarde si vous avez tiré ou non.

POUR JOUER :
Dès que le programme a été lancé par l'ordre RUN, vous devez manœuvrer pour essayer de faire en sorte que l'appareil ennemi soit cadré dans votre viseur.
Pour cela, vous disposez des flèches de déplacement (touche ↑, ↓, →, ←) et vous pouvez tirer à tout moment en pressant la touche Ø.
L'appareil ennemi peut continuer à se déplacer de manière aléatoire, même lorsque vous avez tiré.
La partie se terminera avec l'affichage de GAGNE, si vous touchez l'avion ennemi.
On peut aussi arrêter à tout moment en frappant CNT et C.

EXTENSIONS POSSIBLES :
Il sera assez facile de faire un affichage au graphisme plus soigné, utilisant la couleur. On pourrait envisager de stabiliser la position de l'avion ennemi pendant le tir (il suffit de modifier la ligne 18Ø).
On peut aussi introduire plusieurs avions, compter les points ou le temps, tenir compte de sa propre réserve de carburant, etc.

```
10 REM COMBAT AERIEN
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 A=5:B=15
60 N=21
70 LOCATE B,A,0:COLOR 1,6:PRINT CHR$(127
)+"-"+CHR$(127)
80 U=A:V=B
90 A=A+2*INT(RND*2)-1
100 B=B+2*INT(RND*2)-1
110 A$=INKEY$
120 B=B-2*(A$=CHR$(9))+2*(A$=CHR$(8))
130 A=A-2*(A$=CHR$(11))+2*(A$=CHR$(10))
140 IF B<0 OR B>37 THEN B=ABS(B-37)
150 IF A<3 OR A>22 THEN A=ABS(A-22)
160 LOCATE V,U:PRINT"    "
170 LOCATE 18,9:COLOR 0,6:PRINT "* *":LO
CATE 18,10:PRINT" X ":LOCATE 18,11:PRINT
"* *"
180 IF A$="0" OR N<21 THEN GOSUB 200
190 GOTO 70
200 LOCATE 29-N,N,0:PRINT"*":LOCATE N+9,
N:PRINT"*"
210 LOCATE 29-N,N,0:PRINT" ":LOCATE N+9,
N:PRINT" "
220 N=N-1
230 IF N>9 THEN RETURN
240 IF (A=9 OR A=10 OR A=11) AND (B=18 O
R B=19 OR B=20) THEN PRINT"GAGNE...":PLA
Y"DO;MI;SO":END
250 GOTO 60
```



Dessin de dés

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : lancer de dés.
Difficulté du jeu : facile.

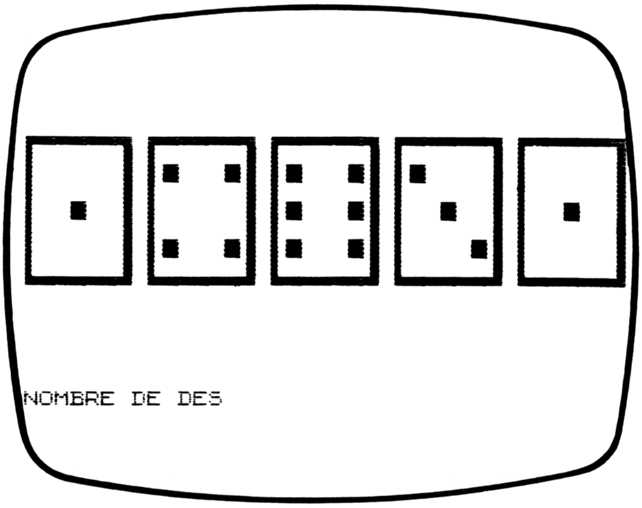
LE JEU :
Le programme proposé est surtout intéressant par son côté graphique puisqu'il permet de représenter, de façon très convenable, de 1 à 5 dés sur l'écran, en utilisant une technique de dessin un peu particulière.
Chaque réponse à la question NOMBRE DE DES provoque immédiatement un nouveau lancer du nombre de dés indiqué.

LE PROGRAMME :
La ligne 6Ø permet de contrôler s'il y a eu ou non action sur une touche de 1 à 5.
Les lignes 9Ø et 1ØØ initialisent les deux variables de dessin des dés.
La boucle I (11Ø-2ØØ) et les lignes 12Ø et 13Ø dessinent le cadre du dé.
La ligne 14Ø réalise le tirage, et la boucle J (15Ø-19Ø) dessine les points en relation avec le tirage réalisé (en appelant les variables A$ et B$).

POUR JOUER :
On lance le programme par l'ordre RUN, et, à la question NOMBRE DE DES, on répond en frappant une touche de 1 à 5. Les dés demandés sont alors représentés, et la question NOMBRE DE DES est de nouveau posée.
Il suffira de taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :
On pourra conserver la partie dessin utilisée dans ce programme pour réaliser toute sorte de jeux de dés, avec une représentation graphique de qualité convenable et facile à modifier si besoin était.

```
10 REM DESSIN DE DES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,4:DEFGR$(0)=15,15,15,15,15,15,
15,15:DEFGR$(1)=240,240,240,240,240,240,
240,240:DEFGR$(2)=0,0,0,0,255,255,255,25
5:DEFGR$(3)=255,255,255,255,0,0,0,0:CLS
50 LOCATE 0,15,0:PRINT"NOMBRE DE DES"
60 A$=INKEY$:IF A$="" THEN 60
70 N=VAL(A$):IF N<1 OR N>5 THEN 40
80 CLS:SCREEN 4,3,0
90 A$="       "+CHR$(127)+"  "+CHR$(127)
+"        "+CHR$(127)+CHR$(127)+"   "+CH
R$(127)
100 B$="121314324515525555"
110 FOR I=1 TO N
120 LOCATE 8*I-8,1:FOR K=1 TO 7:PRINT GR
$(2);:NEXT K
130 LOCATE 8*I-8,9:FOR K=1 TO 7:PRINT GR
$(3);:NEXT K
140 X=INT(RND*6+1)
150 FOR J=0 TO 6
160 IF J<>0 AND J<=3 THEN S=5*VAL(MID$(B
$,3*(X-1)+J,1))-4
170 IF J<>0 AND J<=3 THEN LOCATE 8*I-7,2
*J+1:COLOR 1,3:PRINT MID$(A$,S,5)
180 LOCATE 8*I-8,J+2:COLOR 4,3:PRINT GR$
(1):LOCATE 8*I-2,J+2:PRINT GR$(0)
190 NEXT J
200 NEXT I
210 GOTO 50
```



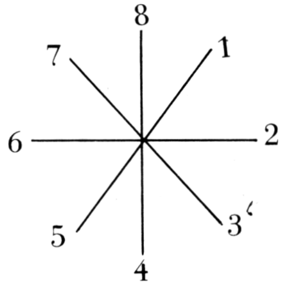
Le jeu du missile

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : contre le MO6.
Difficulté du jeu : moyenne.

LE JEU :
Le MO6 est en alerte : vous êtes aux commandes d'un avion chargé d'une mission d'espionnage sur son territoire ; votre but est de survoler une base ennemie (représentée par un H agrandi), et de revenir sain et sauf sur votre base ; mais, dès que le MO6 vous aura repéré, il va envoyer à votre poursuite, un missile très perfectionné qui n'aura de cesse de vous détruire. Votre avion peut se déplacer dans toutes les directions, grâce aux touches de 1 à 8 ; réussirez-vous à remplir votre mission ?

LE PROGRAMME :
La ligne 5Ø dessine la base ennemie à survoler (sous forme d'un H agrandi).
La ligne 6Ø dessine votre base de départ.
Les lignes 8Ø et 1ØØ choisissent l'instant de départ du missile.
La ligne 12Ø dessine votre avion.
Les lignes 14Ø et 15Ø définissent son mouvement suivant 8 directions, en utilisant les sous-programmes en lignes 1ØØØ, 2ØØØ, 3ØØØ et 4ØØØ.
La ligne 16Ø assure le départ du missile qui est dessiné par la ligne 17Ø et dont le déplacement est géré par les lignes 18Ø et 19Ø.
Les lignes 2ØØ et 21Ø contrôlent la fin de partie.

POUR JOUER :
Lancer le programme par RUN.
Le terrain de jeu s'affiche et vous pouvez déplacer votre appareil en utilisant les touches de 1 à 8, avec la convention suivante :



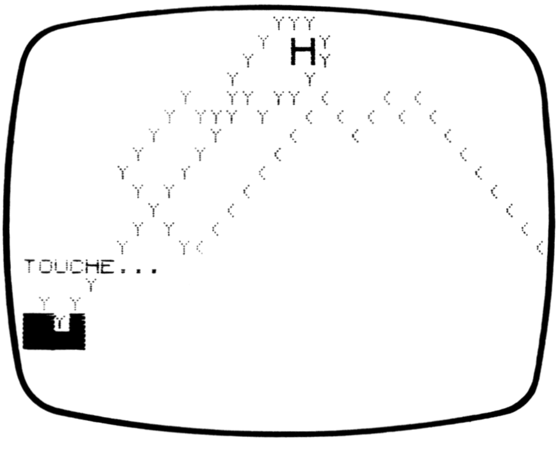
Vous devez lui faire contourner le territoire ennemi (si vous, ou le missile, touchez ce territoire, une deuxième partie commence).
Si le missile vous détruit, vous obtenez un message TOUCHE, et la partie s'arrête.
Taper CNT et C pour arrêter la partie.

EXTENSIONS POSSIBLES :

Il sera sans doute nécessaire de rajouter divers contrôles dans le programme (l'avion a-t-il survolé la base ennemie, est-il revenu à sa propre base...).

On peut aussi prévoir divers niveaux de jeu en faisant, par exemple, varier la vitesse du missile, prévoir l'utilisation de leurres par l'avion, l'explosion du missile au bout d'un certain temps, etc.

```
10 REM LE JEU DU MISSILE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS :M=0
50 LOCATE 17,5:ATTRB 1,1:COLOR 0,6:PRINT
"H"
60 LOCATE 0,20:COLOR 2,6:PRINT CHR$(127)
+CHR$(127)
70 U=15:X=18
80 S=INT(RND*U+6)
90 Y=1
100 S=S-1
110 V=33
120 LOCATE Y,X,0:ATTRB 0,0:COLOR 4,6:PRI
NT"Y"
130 A$=INKEY$:IF A$="" THEN 130
140 F=-2*(VAL(A$)<5)-1
150 ON VAL(A$)+2*(F-1) GOSUB 1000,2000,3
000,4000
160 IF S<>0 THEN 100
170 LOCATE V,U:COLOR 1,6:PRINT "<"
180 U=U-SGN(U-X)
190 V=V-SGN(V-Y)
200 IF U=X AND V=Y THEN PRINT"TOUCHE..."
:PLAY"SO;MI;DO":END
210 IF (U=4 OR U=5) AND (V=17 OR V=18) T
HEN 40
220 IF X<=4 AND Y>= 18 THEN M=1
230 IF M=1 AND X=18 AND Y<=4 THEN PRINT"
MISSION ACCOMPLIE":PLAY"DO;MI;SO":END
240 GOTO 120
1000 X=X-F
2000 Y=Y+F
2010 RETURN
3000 GOSUB 2000
4000 X=X+F
4010 RETURN
```



Le singe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Il s'inspire de l'idée déjà ancienne du singe qui, frappant au hasard sur une machine à écrire, a des chances non nulles de finir par écrire un texte signifiant.
Le rôle du singe est, ici, tenu par le MO6 qui va afficher des groupes de 3 à 6 lettres pendant un temps relativement court.
Si l'un des joueurs estime qu'il peut réussir à faire un mot avec les lettres proposées, il doit taper rapidement sur sa touche de jeu (1 ou Ø).
Le mot devra alors être accepté ou refusé par l'adversaire qui répondra JUSTE ou FAUX.
Chaque mot juste rapporte 1 point, chaque mot faux enlève 1 point.

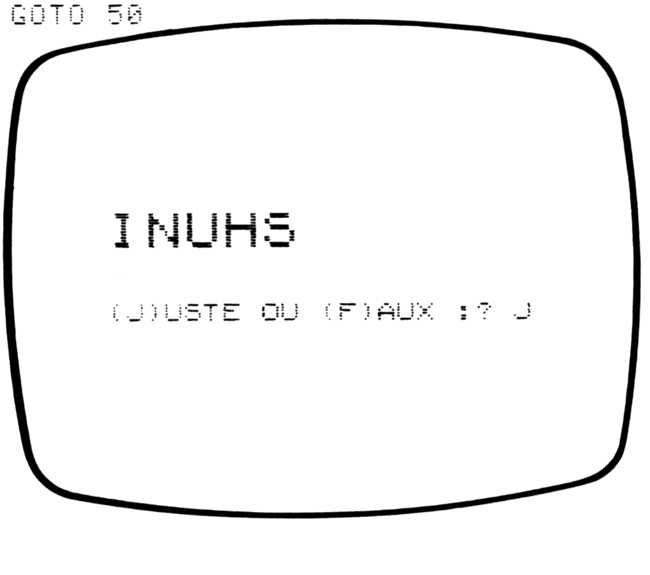
LE PROGRAMME :
La ligne 4Ø contient une liste V$ des voyelles courantes, et une liste C$ des consonnes courantes.
Il sera possible de modifier ces deux listes pour augmenter, par exemple, la fréquence d'apparition de certaines lettres (on pourra ainsi mettre deux E dans V$ ou deux R dans C$, etc.).
La ligne 5Ø définit la longueur du mot (en principe, de 3 à 6 lettres, mais on peut également modifier cette longueur).
La boucle I (8Ø-13Ø) permet la confection d'un "mot" en prenant une lettre, soit dans l'alphabet complet, soit dans la liste C$ (ligne 1ØØ), soit dans la liste V$ (ligne 11Ø).
La ligne 15Ø temporise l'apparition du résultat.
La ligne 17Ø gère la fin de partie, avec affichage du score.
La ligne 18Ø regarde si l'un des joueurs a réagi.
Enfin, les lignes 2ØØ et 21Ø comptabilisent les points de chaque joueur.

POUR JOUER :
On lance le programme en tapant RUN.
La première série de lettres est alors affichée sur l'écran. Le joueur de gauche devra taper 1 s'il voit qu'il peut réaliser un mot, celui de droite devra taper Ø.
Si rien n'a été frappé, une autre série de lettres sera affichée après peu de temps.
Si l'un des joueurs a frappé Ø ou 1, la question (J)USTE ou (F)AUX est alors affichée. On répond J si le mot est correct, F si le mot est incorrect, suivi de ENTREE.
Les points sont alors calculés par MO6, et la série de lettres suivante est alors affichée.

Pour sortir du programme, il faudra taper P au lieu de Ø ou 1 ; le score final est alors affiché.

EXTENSIONS POSSIBLES :
On pourra modifier la longueur des mots, la fréquence d'apparition des lettres, le décompte des points, la règle du jeu, etc.

```
10 REM LE SINGE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 A=0:B=0:V$="AEIOU":C$="BCDGLMNPRSTV"
50 X=INT(RND*4+3)
60 M$=""
70 CLS
80 FOR I=1 TO X
90 E$=CHR$(INT(RND*26+65))
100 IF RND<0.4 THEN E$=MID$(C$,INT(LEN(C
$)*RND+1),1)
110 IF RND<0.5 THEN E$=MID$(V$,INT(LEN(V
$)*RND+1),1)
120 M$=M$+E$
130 NEXT I
140 LOCATE 0,6,0:ATTRB 1,1:COLOR 0,6:PRI
NT M$:PLAY"DO;RE;MI"
150 FOR J=1 TO 500:NEXT J
160 ATTRB 0,0:COLOR 4,6:A$=INKEY$
170 IF A$="P" THEN LOCATE 0,10:PRINT"SCO
RE DE A :";A;" ; SCORE DE B :";B:PLAY"DO
;MI;SO":END
180 IF A$<>"0" AND A$<>"1" THEN 50
190 PRINT:INPUT"(J)USTE OU (F)AUX :";R$
200 A=A-(R$="J" AND A$="0") + (R$="F" AN
D A$="0")
210 B=B-(R$="J" AND A$="1") + (R$="F" AN
D A$="1")
220 GOTO 50
```



Le jeu du 15

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : moyenne.

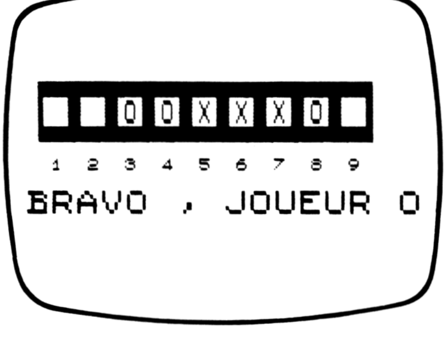
LE JEU :
C'est un jeu pour deux joueurs, assez ancien, qui était autrefois pratiqué dans les foires. On dispose d'un tapis de jeu de 9 cases, et chacun mise à son tour un jeton de sa couleur, sur une case disponible.
Le premier des deux joueurs qui, avec exactement 3 jetons, arrive à faire un total de 15, a gagné.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine le terrain de jeu, et les lignes 9Ø à 1ØØ s'occupent de la gestion du tour des joueurs (numéro 1 ou 2 pour chaque joueur, et jeton O ou X correspondant).
La ligne 11Ø attend le coup joué.
La boucle I (12Ø-14Ø) vérifie la validité du coup.
La ligne 15Ø teste la fin de partie, en cas de match nul, et la ligne 16Ø refuse les coups non conformes (avec retour à la situation initiale, ligne 9Ø).
Les lignes 17Ø et 18Ø s'occupent de l'affichage du coup, et la ligne 19Ø l'enregistre dans une variable R$. La partie la plus intéressante du programme, qui pourrait servir à définir une stratégie machine, va de la ligne 21Ø à la ligne 3ØØ : on utilise 3 boucles emboîtées pour vérifier si l'un des joueurs a obtenu, avec 3 pions, un total de 15. Le changement de joueur s'effectue en ligne 31Ø.

POUR JOUER :
Lancer le jeu en tapant RUN.
Le terrain de jeu s'affiche, puis la question JOUEUR 1 : X, et le joueur qui a choisi les X devra donner le numéro de la case où il désire jouer si elle est libre, son coup y sera enregistré.
On obtient, ensuite, l'affichage JOUEUR 2 : O et le deuxième joueur procède de la même façon. Si l'un des joueurs gagne (total de 15 sur 3 coups), il obtient l'affichage BRAVO et la partie s'arrête.
En cas de match nul, la machine affiche C'EST FINI.
On fera une autre partie en tapant RUN.

EXTENSIONS POSSIBLES :
Le plus intéressant sera sans doute d'essayer de définir une stratégie machine (en essayant d'éviter que la machine ne joue toujours le 5 lorsqu'elle doit commencer).
On peut aussi, bien sûr, améliorer le graphisme et introduire des effets sonores pour les différents messages.

```
10 REM LE JEU DU 15
20 CLS:CLEAR,,4:DEFGR$(0)=0,0,0,0.255,25
5,255,255:DEFGR$(1)=255,255,255,255,0,0,
0,0:DEFGR$(2)=15,15,15,15,15,15,15,15:DE
FGR$(3)=240,240,240,240,240,240,240,240
30 F=0:R$="123456789"
40 O$="":X$=""
50 FOR I=0 TO 24 STEP 3
60 LOCATE I+1,2:COLOR 4,6:ATTRB 0,1:PRIN
T GR$(0)+GR$(0)+GR$(0):LOCATE I+1,4:PRIN
T GR$(3)+" "+GR$(2):LOCATE I+1,6:PRINTGR
$(1)+GR$(1)+GR$(1):ATTRB0,0
70 LOCATE I+1,7:COLOR 0,6:PRINT (I+3)/3
80 NEXT I
90 LOCATE 0,10:PRINT "JOUEUR ";F+1;
100 IF F=1 THEN PRINT" : O" ELSE PRINT"
: X"
110 A$=INKEY$:IF A$<"1" OR A$>"9" THEN 1
10
120 FOR I=1 TO LEN(R$)
130 IF A$=MID$(R$,I,1) THEN GOSUB 170
140 NEXT I
150 IF VAL(R$)=0 THEN LOCATE 0,20:PRINT"
C'EST FINI":PLAY"DO;DO;DO":END
160 GOTO 90
170 IF F=1 THEN O$=O$+A$:LOCATE 3*I-1,4:
COLOR 1,6:ATTRB 0,1:PRINT"O":ATTRB 0,0
180 IF F=0 THEN X$=X$+A$:LOCATE 3*I-1,4:
COLOR 5,6:ATTRB 0,1:PRINT"X":ATTRB 0,0
190 R$=LEFT$(R$,I-1)+"0"+RIGHT$(R$,LEN(R
$)-I)
200 IF F=1 THEN T$=O$ ELSE T$=X$
210 IF LEN(T$)<3 THEN 310
220 FOR I=1 TO 3
230 IF I=LEN(T$) THEN 310
240 FOR J=I+1 TO LEN(T$)
250 IF J=LEN(T$) THEN 310
260 FOR K=J+1 TO LEN(T$)
270 IF VAL(MID$(T$,I,1))+VAL(MID$(T$,J,1
))+VAL(MID$(T$,K,1))=15 THEN 330
280 NEXT K
290 NEXT J
300 NEXT I
310 COLOR 4,6:F=ABS(F-1)
320 RETURN
330 LOCATE 0,10:ATTRB 1,1:COLOR 7,0
340 IF F=1 THEN PRINT"BRAVO , JOUEUR O"
ELSE PRINT"BRAVO , JOUEUR X"
350 PLAY"DO;MI;SO"
```



Adresse

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

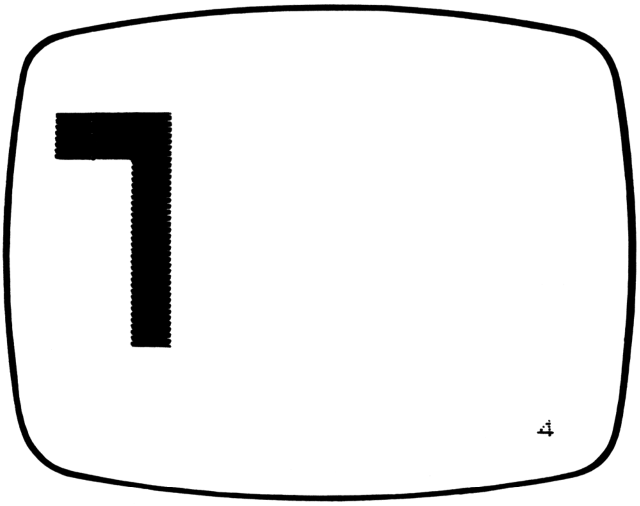
LE JEU :
Des chiffres, entre Ø et 9, vont apparaître au hasard, en divers emplacements de l'écran. Ils restent affichés pendant un temps très court. Le joueur doit réussir à appuyer sur la touche numérique correspondante, au moment où le chiffre est affiché.
Le chiffre tapé par le joueur sera alors affiché en grand format sur la gauche de l'écran ; et chaque réponse exacte rapporte 1Ø points. La durée de la partie est aléatoire et le score sera affiché en fin de partie.

LE PROGRAMME :
Les lignes 5Ø et 6Ø contiennent, dans deux variables alphanumériques C$ et B$, les références qui seront utilisées pour dessiner les chiffres en grand format.
L'emplacement du chiffre tiré par le MO6 est choisi aux lignes 7Ø et 8Ø, et le chiffre est tiré en ligne 9Ø.
La ligne 1ØØ réalise l'impression, et la ligne 11Ø détermine la durée d'affichage. La ligne 13Ø définit la durée du jeu, et la fin de partie est traitée par la ligne 15Ø.
La boucle I (2ØØ-23Ø) réalise le dessin grand format du chiffre frappé par le joueur.
Le contrôle de l'exactitude du résultat est effectué par la ligne 24Ø.

POUR JOUER :
Le programme sera lancé par RUN, puis un premier chiffre apparaît, après une courte pause, vers le milieu de l'écran. Le joueur devra réagir suffisamment vite, tant qu'il reste à l'affichage, et le retaper. Le chiffre qu'il a tapé se dessinera alors en haut à gauche de l'écran, en grand format.
Si le résultat est exact, le joueur marquera 1Ø points.
Le temps de déroulement de la partie sera fonction des chiffres qui apparaîtront à l'affichage.
En fin de partie, le score sera affiché. Il sera possible de s'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager de modifier le jeu pour faire participer plusieurs joueurs, en modifiant éventuellement l'affichage et en codant les touches du clavier mises à la disposition de chacun des joueurs.
On pourra aussi enlever des points à chaque résultat faux, modifier les règles de calcul du score et la durée du jeu, etc.

```
10 REM ADRESSE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 T=100:S=0:C$=CHR$(127):CLS
50 C$=C$+C$+C$+C$+"     "+C$+C$+"  "+C$
60 B$="14441333313121131314413312131121
41133331414114131"
70 X=INT(RND*15+1)
80 Y=INT(RND*15+15)
90 N=INT(RND*10)
100 LOCATE Y,X,0:COLOR 0,1:PRINT N
110 FOR I=0 TO RND*300+150:NEXT I
120 A$=INKEY$:IF A$<>"" THEN 190
130 T=T-N
140 LOCATE Y,X:COLOR 4,6:PRINT"   "
150 IF T>=0 THEN 70
160 CLS
170 PRINT:PRINT"VOTRE SCORE EST DE ";S;"
 SUR 100"
180 PLAY"DO;RE;MI":END
190 U=VAL(A$)
200 FOR I=1 TO 5
210 A=3*VAL(MID$(B$,5*U+I,1))-2
220 LOCATE 2,2*I:COLOR 1,6:ATTRB 1,1:PRI
NT MID$(C$,A,3)
230 NEXT I
240 IF U=N THEN S=S+10
270 GOTO 120
```



Sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : facile.

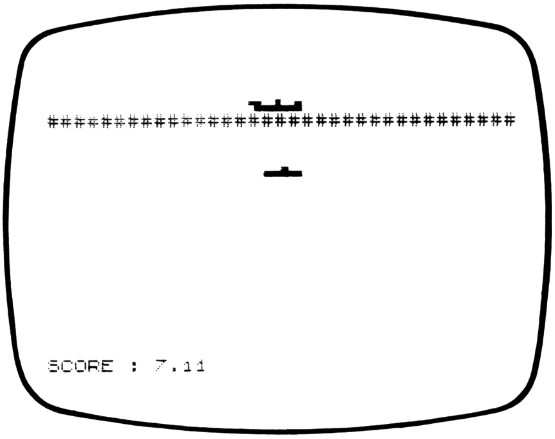
LE JEU :
Vous manœuvrez en surface un chasseur de sous-marins, et des sous-marins passent très régulièrement en dessous de votre unité, à différentes profondeurs.
Vous pouvez, à tout moment, larguer des grenades sous-marines et vous pouvez même envisager de toucher plusieurs fois le même sous-marin.
Votre score sera affiché en permanence, sur la gauche de l'écran (rapport nombre de coups au but sur nombre de sous-marins apparus).

LE PROGRAMME :
La ligne 70 permet au MO6 de choisir la profondeur du sous-marin.
La ligne 90 dessine votre bâtiment.
Les lignes 100 et 110 affichent le score et dessinent la surface de l'eau.
La ligne 120 dessine le sous-marin.
Le tir est effectué par un programme spécifique placé aux lignes 200 à 240.
Le déplacement (très rapide) de votre navire est réalisé par les lignes 170 et 180.
Le test, permettant de savoir si l'on a touché ou non, est effectué en ligne 140.

POUR JOUER :
Taper RUN pour commencer la partie.
Le bâtiment sera déplacé grâce aux flèches droite et gauche.
La touche O autorise le tir.
Le navire, se déplaçant trois fois plus vite que les sous-marins, il sera possible de toucher plusieurs fois le même bâtiment.
Pour terminer la partie, il conviendra de faire CNT et C ou d'attendre 99 sous-marins touchés.

EXTENSIONS POSSIBLES :
Il sera assez élémentaire d'améliorer la qualité graphique du jeu, d'utiliser la couleur et les sons, de faire apparaître plusieurs sous-marins se déplaçant à des vitesses différentes, éventuellement même, de permettre aux sous-marins de tirer...

```
10 REM SOUS-MARIN
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 P=0:C=15:S=5:CLEAR,,5:DEFGR$(0)=0,0,0
,127,127,15,15,15:DEFGR$(1)=0,0,0,0,0,25
5,255,255:DEFGR$(2)=0,0,240,240,240,255,
255,255:DEFGR$(3)=0,0,0,15,15,255,255,25
5:DEFGR$(4)=0,0,0,0,0,63,63,63
50 CLS:COLOR,7:PRINT;SPC(40):PRINT;SPC(4
0):PRINT;SPC(40):COLOR ,6
60 LOCATE 0,S:PRINT"       "
70 T=27:S=INT(RND*8+4):P=P+0.01
80 X=1
90 LOCATE C,2:COLOR 1,7:PRINT "     ";GR$(
0);GR$(1);GR$(2);GR$(3);"    "
100 LOCATE 0,3:COLOR 4,6:PRINT"#########
###############################"
110 LOCATE 0,18:PRINT"SCORE :";P
120 LOCATE T,S,0:COLOR 0,6:PRINT GR$(4);
GR$(3);GR$(1);"  "
130 IF X=1 THEN Y=C+2*X
140 IF S=X AND (T+1=Y OR T+2=Y) THEN P=P
+1:LOCATE T,S:COLOR 1,6:PRINT"***":PLAY"
DO;DO;DO"
150 T=T-1:IF P>99 THEN LOCATE 0,18:ATTRB
 1,1:PRINT"C'EST FINI...":PLAY"DO;RE;MI"
:END
160 IF T<0 THEN 60
170 A$=INKEY$
180 C=C-3*((A$=CHR$(9) AND C<27)-(A$=CHR
$(8) AND C>0))
190 IF A$<>"0" AND X=1 THEN 90
200 LOCATE Y,X:IF X=1 THEN COLOR 0,7 ELS
E COLOR 0,6
210 PRINT" ":X=X+1:COLOR 0,6
220 IF X>S THEN 80
230 LOCATE Y,X:PRINT"O"
240 GOTO 90
```



Zombies

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez facile.

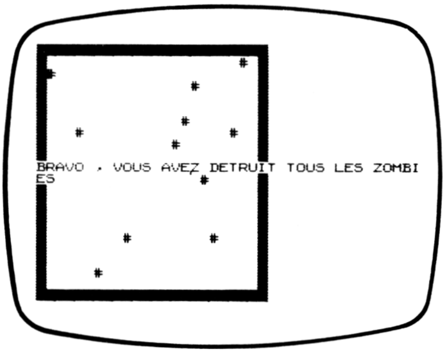
LE JEU :
Les ZOMBIES sont des êtres malfaisants qui vont essayer de vous détruire, mais vous allez, vous aussi, essayer de les utiliser à votre profit, pour débarrasser votre espace vital des trous noirs qui l'obscurcissent. Sur l'écran, est tracé un cadre qui définit votre espace vital. Dans ce cadre, vont apparaître de façon aléatoire, les ZOMBIES (ce sont des Z inversés), les trous noirs (des carrés noirs) et le joueur (un O). Les ZOMBIES convergent vers le joueur pour essayer de le capturer, mais celui-ci se déplace plus rapidement qu'eux, et il va essayer de leur faire avaler tous les trous noirs, sans se faire prendre.
La partie se termine, soit avec la disparition du joueur, soit par l'élimination de tous les ZOMBIES.

LE PROGRAMME :
La position des ZOMBIES est définie par un tableau Z non dimensionné (c'est inutile en dessous de 1Ø).
Les lignes 5Ø à 12Ø (boucle I) dessinent le cadre, les ZOMBIES et les trous noirs.
La ligne 13Ø définit la position initiale du joueur.
Une deuxième boucle I (15Ø-24Ø) réalise le déplacement du joueur.
Les lignes 29Ø à 34Ø contrôlent le déplacement du joueur.

POUR JOUER :
Après avoir lancé l'exécution par RUN, le joueur devra se déplacer en utilisant les 4 flèches et en veillant à ne pas sortir des limites du terrain.
Il devra manœuvrer de façon à faire venir les ZOMBIES sur les trous noirs, sans se faire lui-même coincer.
Pour arrêter la partie avant la fin, il suffira de taper CNT et C.

EXTENSIONS POSSIBLES :
Il serait intéressant d'agrandir le terrain de jeu, d'augmenter le nombre de ZOMBIES ou de trous noirs, d'améliorer les contrôles, de placer le joueur de façon plus aléatoire, éventuellement de modifier la règle.
On pourra aussi, bien sûr, améliorer la qualité du dessin et introduire la couleur.

```
10 REM ZOMBIES
20 S$=INKEY$:Y=RND
30 IF S$="" THEN 20
40 CLS:SCREEN 4,6
50 FOR I=0 TO 21
60 LOCATE I+1,0:PRINT CHR$(127)
70 LOCATE 0,I:PRINT CHR$(127):LOCATE 23,
I:PRINT CHR$(127)
80 IF RND<0.5 THEN LOCATE I,INT(RND*20+1
):COLOR ,0,7:PRINT"#"
90 S=INT(RND*14+1)
100 Z(INT(I/3)+1)=5+S
110 LOCATE I+1,21:COLOR 4,6:PRINT CHR$(1
27)
120 NEXT I
130 K=INT(RND*18+1)
140 P=0
150 FOR I=1 TO 4
160 P=P+Z(I)
170 IF Z(I)=0 THEN 240
180 LOCATE Z(I+4),Z(I):COLOR 4,6:PRINT"
"
190 Z(I)=Z(I)-SGN(Z(I)-S)
200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
210 IF SCREEN(Z(I+4),Z(I))=ASC("#") THEN
 Z(I)=0:GOTO 230
220 LOCATE Z(I+4),Z(I),0:COLOR 1,0:PRINT
"Z"
230 IF Z(I)=S AND Z(I+4)=K THEN PRINT"VO
US AVEZ PERDU...":PLAY"DO;RE;DO":END
240 NEXT I
250 IF P=0 THEN PRINT"BRAVO , VOUS AVEZ
DETRUIT TOUS LES ZOMBIES":PLAY"DO;MI;SO"
:END
260 LOCATE K,S,0:COLOR 1,6:PRINT"O"
270 FOR I=1 TO 1200: NEXT I
280 LOCATE K,S,0:COLOR 4,6:PRINT" "
290 A$=INKEY$
300 K=K-2*((A$=CHR$(9))-(A$=CHR$(8)))
310 IF K<=0 OR K>=23 THEN PRINT"DOMMAGE.
..":PLAY"DO;RE;MI":END
320 S=S-2*((A$=CHR$(10))-(A$=CHR$(11)))
330 IF S<=0 OR S>=21 THEN K=25:GOTO 310
340 GOTO 140
```



Diago

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : difficile.

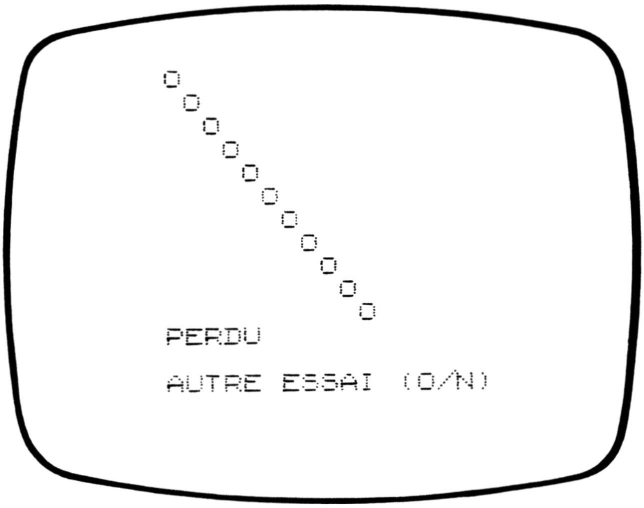
LE JEU :
On lance le programme par RUN et on obtient un premier affichage qui vous renseigne sur la performance à réaliser (par exemple, 5 points noirs ou 17 points noirs...), puis le temps est affiché et décompté, à partir de 1Ø secondes. Le jeu commence lorsque l'on arrive à Ø seconde. Des ronds apparaissent alors en bas et à gauche de l'écran, formant une ligne diagonale, et il vous faudra appuyer sur la touche Ø pour arrêter l'affichage des points dès que vous estimez que le nombre de points demandé au début est apparu à l'écran.
On obtient alors un affichage BRAVO ou PERDU selon le résultat obtenu.

LE PROGRAMME :
Le nombre de points noirs sera choisi au hasard, en ligne 5Ø.
Le temps est décompté par les lignes 6Ø à 12Ø.
Les points sont imprimés et la ligne diagonale est fabriquée par la ligne 14Ø.
La pression de la touche Ø est détectée en ligne 18Ø, et la ligne 19Ø contrôle la fin de partie (aucun réflexe).
Les affichages de fin sont exécutés par les lignes 19Ø à 23Ø.

POUR JOUER :
On lancera l'exécution par RUN.
On obtient alors l'affichage de la phrase LE BON REFLEXE EST A... qui indique le nombre de points gagnants.
Le temps sera ensuite décompté sous la forme ATTENTION : IL RESTE ... SECONDES qui indique le temps restant avant l'apparition de la diagonale.
A l'affichage de Ø, la diagonale apparaît et commence à se déplacer sur l'écran.
Si l'on estime que le nombre de points réclamé a été atteint, on devra presser la touche Ø.
L'affichage indique alors BRAVO si l'on a gagné, ou PERDU dans le cas contraire ; puis la phrase AUTRE ESSAI (O/N) est affichée, répondre O pour faire une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager une adaptation du jeu à deux joueurs ou plus, en laissant la diagonale se former complètement et en testant la coïncidence lors de la pression de la touche affectée au joueur.
On peut aussi réaliser un graphisme plus soigné et modifier le principe du déplacement des objets sur l'écran (déplacement horizontal, vertical ou apparition au hasard...).

```
10 REM DIAGO
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 X=INT(RND*18+3)
60 S=10
70 PRINT"LE BON REFLEXE EST A ";X
80 LOCATE 0,10,0:PRINT"ATTENTION . IL RE
STE ";S;" SECONDES"
90 S=S-1
100 PLAY"DO":FOR I=1 TO 800:NEXT I
110 CLS
120 IF S<>0 THEN 70
130 U=0
140 LOCATE 0,25,0:PRINT TAB(U)"O"
150 FOR I=1 TO 30 : NEXT I
160 U=U+1
170 A$=INKEY$
180 IF A$="0" THEN 210
190 IF U>22 THEN PRINT"PERDU":PLAY"SO;MI
;DO":GOTO 230
200 GOTO 140
210 IF U<>X THEN U=23:GOTO 190
220 PRINT"BRAVO":PLAY"DO;RE;MI"
230 PRINT:PRINT"AUTRE ESSAI (O/N)"
240 A$=INPUT$(1):IF A$="O" THEN 40
```



Décomaque

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solution du programme Nicomaque.
Difficulté du jeu : aucune.

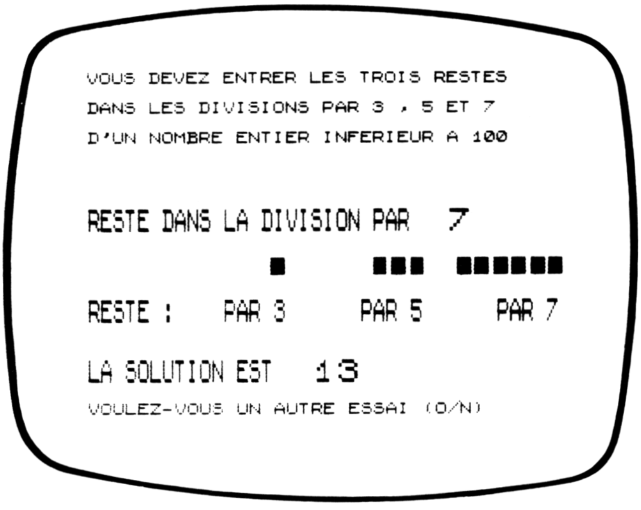
LE JEU :
C'est une réplique au programme Nicomaque, proposé dans les pages précédentes ; vous entrez les restes successivement dans les divisions par 3, 5 et 7 d'un entier compris entre 1 et 100, et le MO6 calcule et affiche cet entier. Le procédé retenu peut d'ailleurs être généralisé (à vous de trouver comment).

LE PROGRAMME :
Les lignes 30 à 50 affichent quelques explications.
La ligne 90 attend que vous frappiez un reste, qui sera refusé s'il est incompatible avec la division par le nombre affiché.
La ligne 100 emmagasine les restes dans une variable non dimensionnée A (S).
La ligne 130 réaffiche les restes entrés.
Les boucles I (140-180) et J (150-170) dessinent des petits carrés selon les restes introduits.
Les lignes 210 à 230 gèrent la fin de partie.

POUR JOUER :
On lance le programme par RUN. Un texte de présentation est affiché, puis le message "RESTE DANS LA DIVISION PAR 3" s'affiche.
On doit répondre 0, 1 ou 2, et l'on obtient le message "RESTE DANS LA DIVISION PAR 5" etc.
La solution est affichée dès que l'on a donné les 3 restes.
Pour une autre partie, taper O à la question VOULEZ-VOUS UN AUTRE ESSAI (O/N).
Tapez N pour finir.

EXTENSIONS POSSIBLES :
On peut étendre le programme aux entiers de 1 à 1000, en utilisant les restes par 3, 5, 7 et 11 ; on pourra rajouter du son et de la couleur pour rendre le programme plus attrayant.

```
10 REM DECOMAQUE
20 CLS:SCREEN 7,0
30 PRINT"VOUS DEVEZ ENTRER LES TROIS RES
TES"
40 PRINT:PRINT"DANS LES DIVISIONS PAR 3
, 5 ET 7"
50 PRINT:PRINT"D'UN NOMBRE ENTIER INFERI
EUR A 100"
60 I=3:S=1
70 LOCATE 0,10:COLOR 0,3:ATTRB 0,1:PLAY"
DO;RE;MI"
80 PRINT"RESTE DANS LA DIVISION PAR ";:C
OLOR 1,7:ATTRB 1,1 :PRINT I
90 A$=INPUT$(1):IF A$<"0" OR A$>CHR$(47+
I) THEN 90
100 A(S)=ASC(A$)-48
110 S=S+1:I=I+2
120 IF I<8 THEN 70
130 LOCATE 0,16:COLOR 4,6:ATTRB 0,1:PRIN
T"RESTE :     PAR 3        PAR 5       PAR 7
"
140 FOR I=1 TO 3
150 FOR J=1 TO A(I)
160 IF A(I)<>0THEN BOXF(40+90*I-12*J,100)
-(48+90*I-12*J ,108),I
170 NEXT J
180 NEXT I
190 T=A(1)*70+A(2)*21+A(3)*15
200 IF T>105 THEN T=T-105:GOTO 200
210 LOCATE 0,20:COLOR 1,6:ATTRB 0,1:PRIN
T"LA SOLUTION EST ";:ATTRB 1,1:COLOR 0,7
:PRINT T
220 LOCATE 0,22:COLOR 4,6:ATTRB 0,0:PRIN
T"VOULEZ-VOUS UN AUTRE ESSAI (O/N)"
230 A$=INPUT$(1):IF A$="O" THEN RUN
```



Chasse au sous-marin

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexion.
Difficulté du jeu : moyenne.

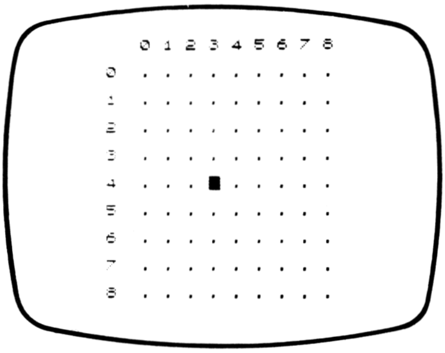
LE JEU :
Vous êtes à bord d'un bâtiment de surface, chargé de traquer un sous-marin caché quelque part dans votre secteur ; vous devez vous déplacer jusqu'à venir au-dessus du sous-marin et tirer à une certaine profondeur ; mais, attention : le sous-marin, lui aussi, est mobile ; il ne coupera ses moteurs que lorsque vous serez exactement à sa verticale, pour ne pas être repéré.

LE PROGRAMME :
La boucle I (6Ø-12Ø) affiche le terrain.
La ligne 14Ø définit la position initiale du sous-marin ; la ligne 15Ø permet son déplacement, contrôlé par la ligne 16Ø (on reste dans les limites du terrain).
La ligne 18Ø affiche votre position.
Les lignes 22Ø à 24Ø contrôlent votre déplacement.
La distance est calculée en ligne 25Ø, affichée en ligne 26Ø et testée en lignes 27Ø et 28Ø.
Le test de fin de partie est effectué à la ligne 31Ø.

POUR JOUER :
On lance le programme par RUN et on entre la dimension ; la grille de jeu s'affiche. Vous êtes au départ en position (Ø, Ø) et vous vous déplacez avec les flèches.
Après chaque déplacement, la nouvelle distance du sous-marin s'affiche ; à chaque coup le sous-marin se déplace. Si vous réussissez à vous placer à sa verticale, la distance Ø est affichée et il ne bouge plus.
Vous tirez en donnant une profondeur entière (entre 2 et 4, bornes comprises). Si vous le ratez, il se replace n'importe où, et on recommence ; si vous le coulez, vous obtenez l'affichage BRAVO et la partie s'arrête.
On peut arrêter à tout moment le programme en utilisant les touches CNT et C.

EXTENSIONS POSSIBLES :
Il est possible de prévoir une amélioration du graphisme, d'améliorer les possibilités de déplacement du chasseur (un peu lent, dans cette version), de prévoir des parcours plus sophistiqués pour le sous-marin, de mettre un compteur de temps...

```
10 REM CHASSE AU SOUS-MARIN
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 PRINT:INPUT"DIMENSION DE LA GRILLE :"
;H
50 CLS:SCREEN 1,0,5
60 FOR I=0 TO H
70 LOCATE 2*I+3,0:PRINT I
80 LOCATE 0,2*I+2:PRINT I
90 FOR J=0 TO H
100 LOCATE 2*J+4,2*I+2:PRINT"."
110 NEXT J
120 NEXT I
130 I=1:X=0:Y=0
140 V=INT(RND*H):U=INT(RND*H)
150 U=U+I:V=V+I
160 IF (U<0 OR U>H OR V<0 OR V>H) THEN I
=-I
170 LOCATE 2*Y+2,2*X+2:IF Y<>0 THEN PRIN
T"."
180 LOCATE 2*Y+4,2*X+2:COLOR 3,0:PRINT C
HR$(127):COLOR 1,0
190 LOCATE 2*Y+6,2*X+2:IF Y<>H THEN PRIN
T"."
200 LOCATE 2*Y+4,2*X:IF X<>0 THEN PRINT"
."
210 LOCATE 2*Y+4,2*X+4,0:IF X<>H THEN PR
INT"."
220 A$=INKEY$:IF A$="" THEN 220
230 Y=Y-(A$=CHR$(9) AND Y<H)+(A$=CHR$(8)
 AND Y>0)
240 X=X-(A$=CHR$(10) AND X<H)+(A$=CHR$(1
1) AND X>0)
250 D=INT(SQR(ABS(X-V)^2+ABS(Y-U)^2))
260 LOCATE 29,10:PRINT"DIST. ";STR$(D)+"
 ";
270 IF D=0 THEN 300
280 IF D<3 THEN 150
290 GOTO 170
300 LOCATE 29,12:INPUT"PROF. ";D
310 IF D=INT(RND*3+2) THEN PRINT"BRAVO":
PLAY"DOMISO":END
320 LOCATE 29,12:PRINT"              ":PLAY"
DODODO"
330 GOTO 140
```



Clavier

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : jeu de réflexe (apprentissage du clavier).
Difficulté du jeu : difficile.

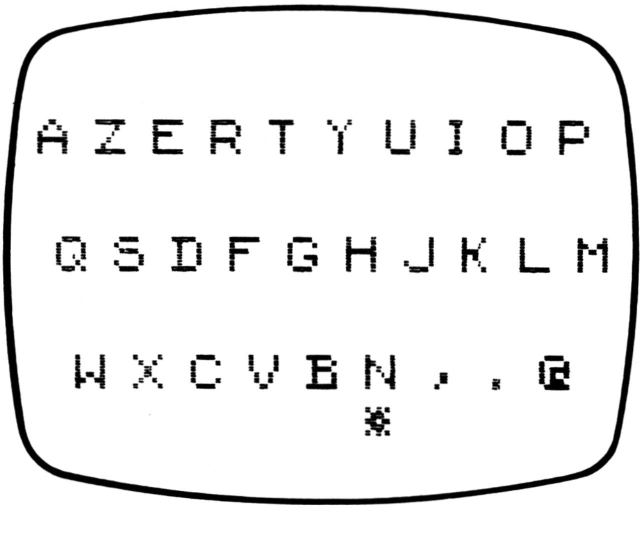
LE JEU :
Une reproduction du clavier de votre ordinateur apparaît sur l'écran ; un point apparaît en dessous d'un caractère, qu'il convient de frapper au clavier. Plus l'on frappe vite, et plus le point se déplace rapidement (le temps écoulé depuis le début s'affiche en fin de partie).

LE PROGRAMME :
Une variable A$ dimensionnée en ligne 6Ø, contiendra la liste des touches concernées. Les valeurs des touches sont entrées de la ligne 7Ø à la ligne 12Ø par appel d'un sous-programme situé en 3ØØ.
Les boucles emboîtées J (13Ø-17Ø) et I (14Ø-16Ø) dessinent le clavier.
La ligne 18Ø définit la position du point imprimé par la ligne 19Ø.
La touche frappée est testée aux lignes 22Ø-23Ø et le point est effacé par la ligne 25Ø.
Les affichages de fin sont réalisés aux lignes 27Ø à 29Ø.

POUR JOUER :
On lance le programme par RUN. L'on dispose alors de quelques secondes pour frapper la touche qui occupe la colonne au-dessus de l'étoile, puis une autre étoile est affichée, etc.
La durée de la partie n'est pas limitée ; il suffit de taper Ø pour obtenir l'affichage du score, et du temps total.
Faire RUN pour une autre partie.

EXTENSIONS POSSIBLES :
On pourra envisager de faire clignoter les lettres, ou de les afficher en couleur, au lieu d'indiquer la colonne par une balle. On pourra afficher les lettres devinées justes en bas de l'écran, et éventuellement s'entraîner à taper un texte qui défile en haut de l'écran, lettre par lettre.

```
10 REM CLAVIER
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 T=0:S=0:P=0
60 DIM A$(10,3)
70 S$="AZERTYUIOP":J=1
80 GOSUB 300
90 S$="QSDFGHJKLM":J=2
100 GOSUB 300
110 S$="WXCVBN,.@ ":J=3
120 GOSUB 300
130 FOR J=1 TO 3
140 FOR I=1 TO 10
150 LOCATE 3*I+J,5*J:ATTRB 1,1:COLOR 1,3:PRINT A$(I,J)
160 NEXT I
170 NEXT J
180 X=INT(RND*10+1):Y=INT(RND*3+1)
190 LOCATE 3*X+Y,2+5*Y,0:COLOR 0,6:PRINT "*"
200 FOR F=1 TO 50+RND*80:A$=INKEY$
210 T=T+1:IF A$="" THEN NEXT F
220 IF A$=A$(X,Y) THEN S=S+1
230 IF A$="0" THEN 270
240 P=P+1
250 LOCATE 3*X+Y,2+5*Y:PRINT" "
260 GOTO 180
270 LOCATE 0,20:ATTRB 0,0:PRINT S;" TROUVES JUSTES SUR ";P;" ESSAIS"
280 PRINT"VOTRE TEMPS :";T
290 PLAY"DOMISO":END
300 FOR I=1 TO 10 : A$(I,J)=MID$(S$,I,1):NEXT I:RETURN
```



Le lézard d'Alice

REFERENCES :
Difficulté de programmation : facile.
Catégorie de jeu : solitaire.
Difficulté du jeu : assez difficile.

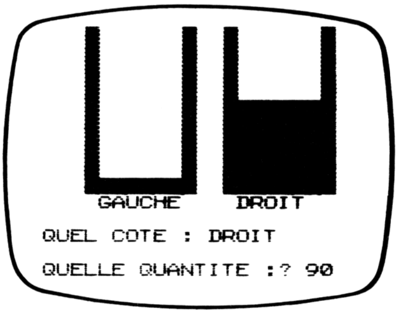
LE JEU :
Il s'inspire très vaguement d'une idée de LEWIS CARROLL, dans *Alice au Pays des Merveilles*. Dans l'histoire originale, Alice grandissait ou diminuait, alors que dans le jeu, pour des raisons d'encombrement de la mémoire, c'est un lézard qui change de taille.
Deux verres apparaissent sur l'écran : l'un contient le liquide qui fait grandir, l'autre le liquide qui fait rapetisser (malheureusement ce n'est pas toujours le même verre). Le joueur devra indiquer la quantité de liquide qu'il désire boire et le lézard changera de taille en fonction de cette quantité. Le but du jeu est de parvenir à lui rendre sa taille initiale.

LE PROGRAMME :
La boucle I (6Ø-9Ø) dessine les verres.
Les deux boucles emboîtées I (18Ø-22Ø) et J (19Ø-21Ø) les remplissent avec la quantité indiquée en ligne 17Ø.
La variation de taille du lézard est définie par la ligne 26Ø, avec contrôle par les lignes 27Ø et 28Ø.
La boucle I (29Ø-31Ø) dessine le corps du lézard, les lignes 32Ø à 37Ø dessinent l'œil et les pattes.
Un message gagnant sera affiché par la ligne 39Ø si l'on a trouvé la bonne taille.

POUR JOUER :
On lance le programme par RUN : les deux verres se dessinent, puis apparaît la question QUEL COTE à laquelle on répond en tapant G ou D ; on obtient l'affichage QUELLE QUANTITE ; on entre alors une quantité (de Ø à 13Ø) et on frappe ENTREE.
Le verre désigné se remplit sous vos yeux, puis, après quelques instants, le lézard est affiché. On devra alors frapper une touche quelconque ; si le lézard n'avait pas la bonne taille, aucun message n'est affiché et l'on revient au dessin des verres.

EXTENSIONS POSSIBLES :
Puisqu'il s'agit d'un jeu essentiellement graphique, il sera intéressant d'améliorer les dessins, d'introduire la couleur, de modifier le caractère aléatoire du choix des verres, etc.
Pourquoi ne pas relire *Alice au Pays des Merveilles*, les idées n'y manquent pas.

```
10 REM LE LEZARD D'ALICE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 0,7
50 N=18
60 FOR I=1 TO 9
70 LOCATE 3,I+4:PRINT CHR$(127):LOCATE 1
0,I+4:PRINT CHR$(127)
80 LOCATE 13,I+4:PRINT CHR$(127):LOCATE
20,I+4:PRINT CHR$(127)
90 NEXT I
100 FOR I=3 TO 10:LOCATE I,14:PRINT CHR$
(127):NEXT I
110 FOR I=13 TO 20:LOCATE I,14:PRINT CHR
$(127):NEXT I
120 LOCATE 4,15:PRINT"GAUCHE":LOCATE 14,
15:PRINT"DROIT"
130 LOCATE 0,17:PRINT "QUEL COTE :"
140 A$=INKEY$:IF A$<>"D" AND A$<>"G" THE
N 140
150 U=2-20*(A$="D")
160 LOCATE 12,17:IF A$="D" THEN PRINT"DR
OIT" ELSE PRINT"GAUCHE"
170 PRINT:INPUT"QUELLE QUANTITE :";Q
180 FOR I=1 TO 4*SQR(Q)
190 FOR J=0 TO 47
200 PSET (4*U+J+24,112-I),4
210 NEXT J
220 NEXT I:PLAY"DOREMI"
230 A$=INKEY$:IF A$="" THEN 230
240 CLS:SCREEN 1,0
250 H=2*(RND<0.5)-1
260 N=N+H*INT(SQR(Q))
270 IF N<12 THEN N=12
280 IF N>60 THEN N=60
290 FOR I=0 TO 4*(N+3)
300 PSET(I,100),1:PSET(I,99),1:PSET(I,98
),1:PSET(I,97),1
310 NEXT I
320 T=4*(N-INT((SQR(N)/2)))
330 PSET(T,88),4:PSET(T-1,89),4:PSET(T-2
,90),4:PSET(T+1,89),4:PSET(T+2,90),4
340 T=4*SQR(N):PSET(T,102),4:PSET(T-1,10
3),4:PSET(T-2,104),4
350 T=T+2:PSET(T,102),4:PSET(T+1,103),4:
PSET(T+2,104),4
360 T=4*(N-SQR(N)):PSET(T,102),4:PSET(T-
1,103),4:PSET(T-2,104),4
370 T=T+2:PSET(T,102),4:PSET(T+1,103),4:
PSET(T+2,104),4
380 A$=INKEY$:IF A$="" THEN 380
390 IF N=28 THEN PRINT"BRAVO":PLAY"DOMIS
O":END
400 GOTO 40
```



Calculatrice

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : simulation d'une calculatrice.
Difficulté du jeu : moyenne.

LE JEU :
Il s'agit de la simulation, en affichage grand format, du fonctionnement d'une petite calculatrice de poche, de type 4 opérations. Les applications pédagogiques de ce programme sont évidentes et son fonctionnement très simple à maîtriser.

LE PROGRAMME :
Les variables B$ et C$, des lignes 2Ø et 3Ø, contiennent les informations codées qui permettront le dessin des chiffres en grand format.
Les lignes 6Ø à 15Ø s'occupent de la gestion des entrées.
La boucle I (16Ø-19Ø) réalise la représentation, en grand format, des chiffres sur l'écran, au fur et à mesure des entrées.
Le point décimal et le signe – sont gérés par les lignes 2ØØ et 21Ø.
Le nombre total des symboles entrés est contrôlé par la ligne 22Ø.
Le résultat est inscrit grâce à un programme fonctionnant en automatique qui se trouve aux lignes 27Ø à 39Ø (l'automatisme est géré par la ligne 23Ø, dans le programme principal).
Les sous-programmes placés aux lignes 4ØØ-41Ø, 42Ø-43Ø, 44Ø-45Ø et 46Ø-47Ø effectuent l'opération indiquée, selon la valeur de Z (ligne 31Ø).
Chaque résultat est mémorisé en A (F-1) (ligne 33Ø).

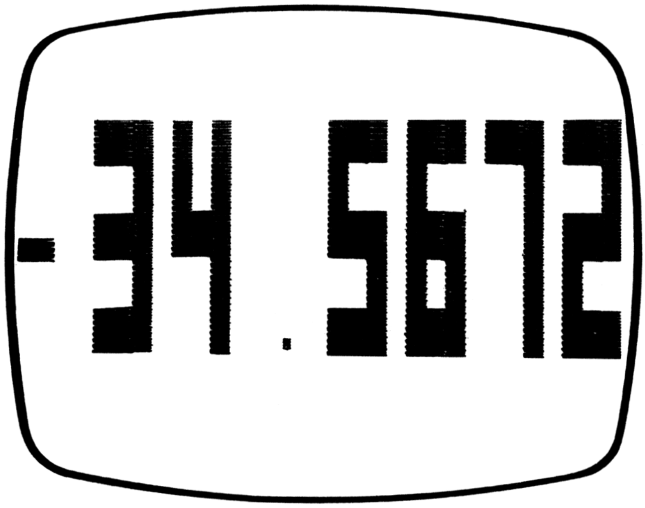
POUR JOUER :
On devra lancer le programme par RUN.
Puis, on tape les chiffres du premier nombre à entrer qui vont s'afficher un par un, au fur et à mesure (on ne pourra entrer plus de 8 symboles, y compris le signe – et le point décimal). La frappe d'un signe opératoire laisse l'affichage intact, mais l'introduction du prochain chiffre ou symbole provoque l'effacement immédiat et l'affichage du nouveau symbole.
Il est à remarquer que les chiffres s'introduisent de gauche à droite, sens de l'écriture usuelle.
L'appareil accepte, comme signes opératoires les symboles +, –, *, / ; le signe = provoquera l'affichage du résultat. Les opérations sont enchaînées automatiquement, le résultat précédent étant pris en compte comme opérateur du calcul suivant.
Pour faire une autre opération, il faudra taper CNT et C, puis RUN.
Taper CNT et C pour sortir du programme.

EXTENSIONS POSSIBLES :
On devra, tout d'abord, améliorer l'affichage du signe – (représenté, ici, par un trait), puis on pourra améliorer le dessin des chiffres (en utilisant, par exemple, le générateur du MO6), on peut aussi envisager une extension à plus de 8 chiffres, autoriser des opérations plus complexes (COS, SIN, ...), ajouter des mémoires, etc.

```
10 REM CALCULATRICE
20 C$=CHR$(127):C$=C$+C$+C$+C$+"    "+C$
+C$+" "+C$
30 B$="144413333313121131314413312131121
41133331414114131"
40 F=1:L=0:CLS:SCREEN 1,6:ATTRB 0,1
50 N=0:T=0:K=0:R$=""
60 A$=INKEY$:IF A$="" THEN 60
70 IF K=1 THEN N=0:R$="":K=0
80 IF A$="=" THEN 260
90 X=(1 AND A$="+")+(2 AND A$="-")+(3 AN
D A$="*")+(4 AND A$="/")
100 R$=R$+A$
110 IF X<>0 THEN 260
120 IF R$="-" THEN A$=R$:GOTO 200
130 IF A$<"." OR A$>"9" OR A$="/" THEN 6
0
140 IF N=0 THEN CLS
150 IF A$="." OR A$="-" THEN 200
160 FOR I=1 TO 5
170 S=3*VAL(MID$(B$,I+5*VAL(A$),1))-2
180 LOCATE N,2*I+3,0:PRINT MID$(C$,S,3)
190 NEXT I
200 IF A$="-" THEN CLS:LOCATE 0,9:ATTRB
1,0:PRINT CHR$(127):ATTRB 0,1
210 IF A$="." THEN LOCATE N+1,13:ATTRB 1
,1:PRINT".":ATTRB 0,1
220 N=N-4*(N<28)
230 IF T<>0 THEN T=T+1:IF T<=LEN(R$) THE
N 380
240 IF K=1 AND L<>1 THEN 50
250 T=0:GOTO 60
260 IF R$="-" AND A(F)=0 THEN 120
270 N=0:K=1
280 IF(A$="*" OR A$="/") AND VAL(R$)=0 T
HEN A(F)=1: GOTO 300
290 A(F)=VAL(R$)
300 IF X<>0 THEN Z=X
310 ON Z GOSUB 400,420,440,460
320 IF A$<>"=" THEN F=F+1 ELSE F=2
330 A(F-1)=VAL(R$)
340 IF VAL(R$)=0 THEN A(F-1)=A(F)
350 R$=STR$(VAL(R$))
360 IF LEFT$(R$,1)=" " THEN R$=RIGHT$(R$
,LEN(R$)-1)
370 T=1
```

→

```
380 A$=MID$(R$,T,1)
390 GOTO 140
400 R$=STR$(A(F-1)+A(F)):A(F)=VAL(R$):L=
1
410 RETURN
420 IF F<>1 THEN R$=STR$(A(F-1)-A(F)):A(
F)=VAL(R$):L=1
430 RETURN
440 IF F<>1 THEN R$=STR$(A(F-1)*A(F)):A(
F)=VAL(R$):L=1
450 RETURN
460 IF F<>1 THEN R$=STR$(A(F-1)/A(F)):A(
F)=VAL(R$):L=1
470 RETURN
```



Niveau 5

Nom du programme	*Thèmes et particularités*
Smokey	Fumez sans risque, avec ce programme original pour un joueur.
Course de voitures	Un jeu de réflexe pas trop difficile, pour un joueur.
Danger spatial	Serez-vous assez adroit pour éviter toutes les météorites ?
Tic Tac Toe	Un classique, mais pas sans intérêt (deux joueurs).
Anagrammes	Un bon jeu de réflexe : pour deux joueurs.
Oeufs	Ne laissez pas faire la pondeuse diabolique (un joueur).
Taquin de lettres	Un proverbe est caché dedans : à vous de le découvrir.
Cibles	Un peu d'adresse, beaucoup de chance : très visuel.
Gouttes	Un véritable gag visuel. A exploiter.
Obstacles	Beaucoup d'obstacles à éviter dans ce jeu d'adresse pour un joueur.
Captures dans l'espace	Votre mission serait facile, sans ces damnées météorites.
Anomalie	Exercez votre perspicacité dans la reconnaissance des symboles.
Plusieurs en un	Un terrain de jeu et les pions, pour deux joueurs.
Parcours du cavalier	Très spectaculaire, et instructif à plus d'un titre.
Mur de briques	Un grand classique dans une version très complète.
Othello	Pour deux joueurs ; il retourne les pions à votre place.
Dames	Pour deux joueurs : classique.
Horloge 2	Utilisation du crayon optique de l'ordinateur.
Quatre à quatre	Un jeu de réflexion un peu difficile, pour deux joueurs.
Echecs	Un programme assez simplifié, pour deux joueurs.

Techniques particulières

LA RECONNAISSANCE D'OBJETS SUR L'ECRAN : LECTURE D'UN CARACTERE PLACE EN LIGNE Y, COLONNE X, PAR SCREEN (X, Y).

Le MO6 dispose d'une instruction spécifique, très pratique pour de nombreux jeux, permettant de reconnaître le code ASCII du caractère présent sur l'écran à l'intersection de la colonne X et de la ligne Y : ceci permettra d'identifier facilement ce caractère ; il suffit d'écrire, par exemple : **C = SCREEN (X, Y)** et C contient le code ASCII du caractère présent.

Bien entendu, ce procédé ne fonctionne ni avec les caractères agrandis, ni avec les caractères graphiques prédéfinis par l'utilisateur (GR$).

On peut cependant s'en sortir dans ce cas, en utilisant une instruction plus difficile à manipuler, qui donne la couleur d'un point de l'écran, ainsi qu'un signe permettant de savoir s'il appartient au graphisme ou au fond : on écrit **POINT (X, Y)** pour avoir la couleur du point se trouvant en X (coordonnée horizontale, de Ø à 319) et en Y (coordonnée verticale, de Ø à 199).

Le nombre renvoyé va de Ø à 15 si le point appartient au graphisme, et de −1 à −16 si le point appartient au fond.

LE CRAYON OPTIQUE

Le crayon optique est l'un des attraits du micro-ordinateur MO6 ; il permet la lecture directe des coordonnées d'un point de l'écran, et l'écriture éventuelle.

Beaucoup des jeux de ce livre pourront être modifiés dans le sens d'une plus grande facilité d'utilisation, à l'aide de ce crayon. Le programme Horloge de ce chapitre vous en fournit un exemple, que vous pourrez facilement adapter vous-même à d'autres jeux.

Smokey

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

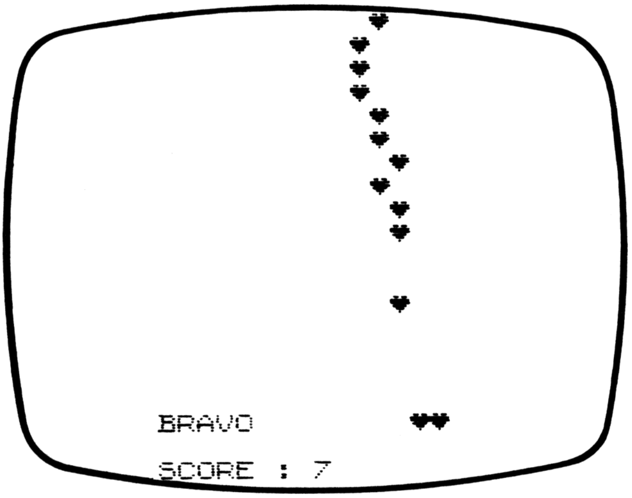
LE JEU :
Une colonne de fumée grise apparaît sur l'écran. Le joueur, représenté par un point noir, devient visible sur la droite. Il doit dans un premier temps se positionner sur la colonne de fumée, puis à partir de là, il va automatiquement avaler la fumée, à condition qu'il réussisse à se maintenir sur sa trace.

LE PROGRAMME :
La ligne 90 fait apparaître la fumée au bas de l'écran ; la ligne 100 permet de changer (vers la droite ou vers la gauche) la direction de la colonne, et la ligne 120 fait monter la colonne.
Le joueur apparaît grâce à la ligne 140, et son déplacement est effectué par les lignes 180 et 190.
Les lignes 250 et 260 se chargent de l'affichage du résultat.

POUR JOUER :
On doit lancer l'exécution du programme par RUN.
La fumée commence à s'élever, tandis que le joueur, apparu sous forme d'un rond coloré dans un coin de l'écran, va se déplacer sur elle à l'aide des 4 flèches. Dès qu'il arrive sur la fumée, il va automatiquement commencer à l'avaler, et devra essayer de suivre la colonne sans dévier (grâce aux flèches). Il gagne s'il arrive en bas de l'écran avant que la fumée n'arrive en haut.
L'affichage final indique alors BRAVO ou PERDU, ainsi que le score obtenu.

EXTENSIONS POSSIBLES :
On pourra, tout en conservant l'idée de départ, imposer d'autres règles : par exemple avaler toute la fumée, qui pourra aussi repartir d'un autre endroit si l'on dévie de la colonne initiale.
On pourra aussi améliorer les contrôles de fin de partie, et la qualité du graphisme.

```
10 REM SMOKEY
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,2:DEFGR$(0)=0,28,62,127,127,12
7,62,28
50 DEFGR$(1)=0,54,127,127,62,28,8,0
60 S=0:X=24:Y=INT(RND*24)
70 U=0:V=15
80 CLS
90 LOCATE Y,X:COLOR 1,6:PRINT GR$(1)
100 Y=Y-(RND<0.5 AND Y<24) + (RND<0.5 AN
D Y>0)
110 T=1
120 X=X-T
130 IF X=0 THEN 250
140 LOCATE V,U,0:COLOR 0,6:PRINT GR$(0)
150 IF T<>0 THEN FOR I=1 TO 300:A$=INKEY
$:IF A$<>"" THEN 170
160 IF T<>0 THEN NEXT I
170 LOCATE V,U:PRINT" "
180 U=U-(A$=CHR$(10) AND T=1 AND X<24)
190 U=U-(T=0)
200 V=V-(A$=CHR$(9) AND V<24)+(A$=CHR$(8
) AND V>0)
210 C=POINT(8*V+4,8*U+4)
220 IF C<>1 THEN 90
230 T=0:S=S+1
240 IF U<23 THEN 140
250 IF T=0 THEN PRINT"BRAVO":PLAY"DOMISO
":GOTO 270
260 PRINT"PERDU...":PLAY"DODODO"
270 PRINT:PRINT"SCORE :";S
```



Course de voitures

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (1 joueur).
Difficulté du jeu : moyenne.

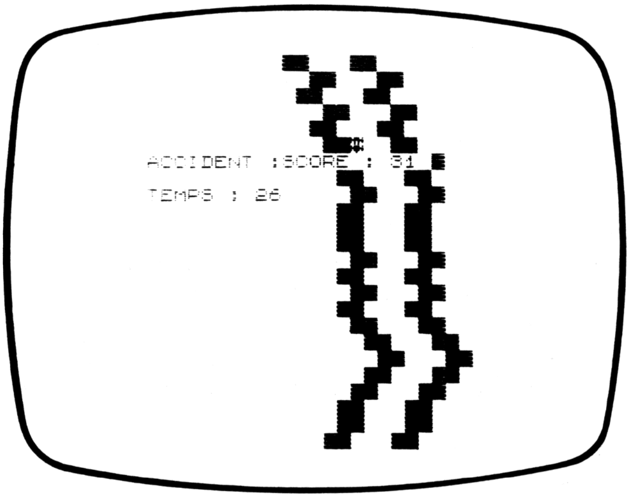
LE JEU :
Vous circulez sur une route très sinueuse, ne comportant aucun obstacle, et vous devez essayer de la suivre le plus longtemps possible. La trace de votre parcours reste affichée en permanence, et votre score est affiché dès que vous avez un accident.

LE PROGRAMME :
L'impression de la route est assurée par la boucle I (7Ø-1ØØ) et la ligne 9Ø imprime la couleur de la route.
La ligne 11Ø fait avancer la route, et la ligne 12Ø imprime la voiture.
La ligne 15Ø contrôle son mouvement, et la ligne 16Ø détermine si la route fait ou non des virages.
La ligne 17Ø regarde si l'on a touché le bord de la route.
La ligne 18Ø incrémente le score, et la ligne 19Ø efface l'ancienne position de la voiture (une trace claire apparaît à l'emplacement de sa trajectoire).
La ligne 22Ø affiche le temps avant l'accident.

POUR JOUER :
Lancer l'exécution du programme par l'ordre RUN, le dessin de la route commence à évoluer, et votre véhicule, représenté par une voiture, apparaît.
Pour le déplacer vers la droite ou vers la gauche, vous devrez utiliser les touches → et ← (flèches de déplacement du curseur). Il vous faudra éviter le plus longtemps possible de toucher une bordure noire, ce qui provoquera l'arrêt du programme et l'affichage de la mention ACCIDENT, suivie de l'indication sur le score réalisé.
On pourra arrêter le déroulement de la partie à tout moment en pressant les touches CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager de jouer à plusieurs, chacun son tour, en enregistrant le meilleur score.
On peut aussi envisager d'élargir la piste, et de faire courir plusieurs voitures.
On pourra aussi s'inspirer des jeux d'arcades qui sont particulièrement nombreux sur ce thème.

```
10 REM COURSE DE VOITURES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,1:DEFGR$(0)=93,127,85,20,93,12
7,93,8
50 T=0:CLS
60 X=24:Y=11:A=5:B=10:S=0
70 FOR I=0 TO 6
80 LOCATE Y+I,X:COLOR 7,6:PRINT CHR$(127
);
90 IF I<2 OR I>4 THEN LOCATE Y+I,X:COLOR
 0,6:PRINT CHR$(127);
100 NEXT I
110 PRINT
120 LOCATE B,A,0:COLOR 4,6:PRINT GR$(0)
130 FOR I=1 TO 10:A$=INKEY$:T=T+0.1:IF A
$="" THEN NEXT I
140 C=B
150 B=B-(A$=CHR$(9))+(A$=CHR$(8))
160 Y=Y-(-2*RND AND Y<25)+(-2*RND AND Y>3)
170 IF POINT(8*B+3,8*A+11)=0 THEN 210
180 S=S+1
190 LOCATE C,A:COLOR ,6:PRINT" "
200 GOTO 70
210 PRINT"ACCIDENT :SCORE :";S
220 PRINT:PRINT"TEMPS :";INT(T):PLAY"DOR
EMI"
```



Danger spatial

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe, pour un joueur.
Difficulté du jeu : moyenne.

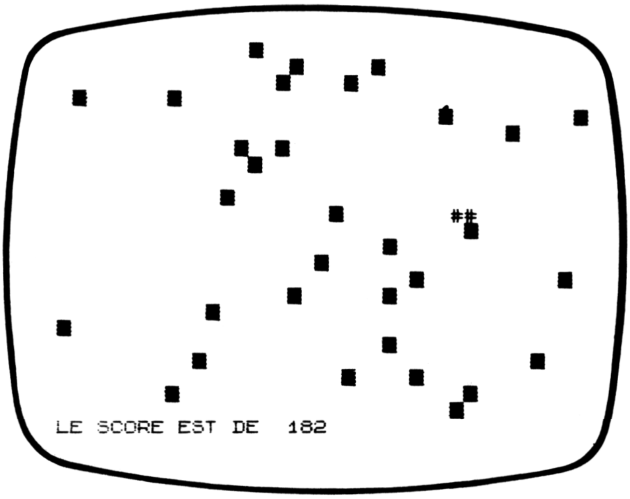
LE JEU :
Vous êtes aux commandes d'un engin spatial qui vient de pénétrer dans un nuage de météorites. Vous devez essayer de franchir le nuage sans heurter une seule météorite. Votre score sera affiché en fin de partie, lorsque vous entrez en collision avec une météorite.

LE PROGRAMME :
La position de chaque météorite est déterminée par les lignes 6Ø et 8Ø ; elle est imprimée par les lignes 7Ø et 9Ø.
L'avance des météorites est réalisée par la ligne 1ØØ.
Les lignes 1ØØ et 11Ø vérifient si vous avez ou non heurté une météorite.
La ligne 13Ø dessine votre vaisseau spatial, et la ligne 16Ø permet de le déplacer.
Le compteur se trouve en ligne 17Ø.

POUR JOUER :
Le jeu est lancé par l'ordre RUN.
Votre vaisseau apparaît vers le milieu de l'écran, et les météorites viennent à votre rencontre en partant du bas de l'écran. Pour vous déplacer, vous pouvez utiliser les touches de déplacement horizontal (→ et ←).
On peut arrêter la partie à tout moment en tapant CNT et C. Sinon, le jeu s'arrête lorsque vous heurtez une météorite, et le score est alors affiché.

EXTENSIONS POSSIBLES :
Il serait intéressant de rendre le jeu de plus en plus difficile en faisant apparaître de plus en plus de météorites au fur et à mesure du déroulement de la partie.
On peut aussi essayer de détruire les météorites en les pulvérisant au fur et à mesure de leur apparition.

```
10 REM DANGER SPATIAL
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 7,0
50 X=18:Z=0:Y=X
60 H=INT(RND*40)
70 IF RND<0.7 THEN LOCATE H,24:COLOR 1:P
RINT CHR$(127);
80 K=INT(RND*40)
90 IF RND<0.6 THEN LOCATE K,24:COLOR 1:P
RINT CHR$(127);
100 PRINT:B=SCREEN(X,12):C=SCREEN(X+1,12
)
110 IF B=127 OR C=127 THEN 190
120 LOCATE Y-1,11:COLOR 0:PRINT"    ":LO
CATE Y-1,10:PRINT"    "
130 LOCATE X,12,0:COLOR 6:PRINT"##"
140 A$=INKEY$
150 Y=X
160 X=X-(A$=CHR$(9) AND X<38)+(A$=CHR$(8
) AND X>0)
170 Z=Z+1
180 GOTO 60
190 PRINT"LE SCORE EST DE ";Z:PLAY"DOMIS
O"
```



Tic tac toe

REFERENCES :
Difficulté de programmation : moyenne.
Catégorie de jeu : réflexion (deux joueurs).
Difficulté du jeu : facile.

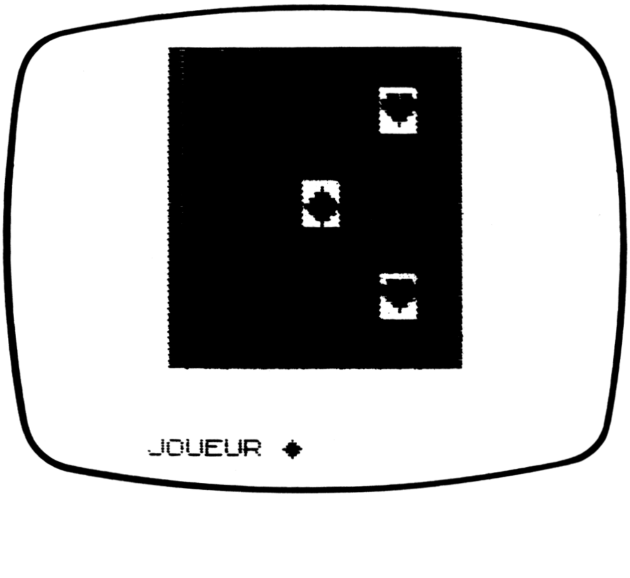
LE JEU :
Un grand classique des jeux d'écoliers, proposé ici dans une version ne comportant pas le jeu machine ; chaque joueur devra placer à son tour un Ø ou un X (ici un cœur ou un carreau) dans une case libre de la grille de jeu, et le premier qui réussit un alignement de 3 symboles identiques a gagné (l'alignement peut être horizontal, vertical, ou suivant l'une des deux diagonales).

LE PROGRAMME :
Le terrain de jeu est dessiné à l'aide d'une boucle I (4Ø-8Ø).
La ligne 1ØØ s'occupe d'afficher le symbole cœur ou carreau correspondant au joueur qui a le trait.
Un symbole clignotant (ligne 13Ø) est affiché en haut et à gauche du terrain, même si la case est déjà occupée ; les lignes 16Ø et 17Ø permettent de déplacer le symbole sur le terrain de jeu.
La ligne 18Ø vérifie si la case où l'on veut jouer est libre, et si l'on désire jouer.
La ligne 19Ø affiche le coup, et la ligne 2ØØ réalise le changement de joueur.

POUR JOUER :
Le terrain de jeu sera affiché dès la frappe de RUN.
Un symbole clignotant apparaît alors en haut et à gauche de l'écran ; le joueur peut le déplacer sur le terrain de jeu à l'aide des 4 flèches (touches ←, → et ↑, ↓).
Lorsqu'il arrive sur la case où il désire jouer, il suffit qu'il appuie sur la touche Ø, le coup sera refusé si la case n'est pas libre.
Si la case est libre, le symbole qui correspond à ce joueur est inscrit, et on change de joueur.
Lorsque la partie est finie, il convient de taper CNT et C pour sortir du programme. On pourra ensuite faire une autre partie avec RUN.

EXTENSIONS POSSIBLES :
On pourrait réaliser après chaque coup un contrôle pour vérifier si l'un des joueurs a gagné. On peut aussi réaliser un jeu machine mais il faudrait éviter qu'il soit par trop systématique (par exemple, la machine jouant toujours son premier coup sur la case centrale).

```
10 REM TIC TAC TOE
20 CLS:SCREEN4,6:CLEAR,,2:DEFGR$(0)=0,54
,127,127,62,28,8,0:DEFGR$(1)=0,8,28,62,1
27,62,28,8
30 BOXF(10,10)-(130,120),5
40 FOR I=1 TO 3
50 FOR J=1 TO 3
60 ATTRB 1,1:LOCATE 4*I,4*J:COLOR 6:PRIN
T CHR$(127)
70 NEXT J
80 NEXT I
90 K=0
100 LOCATE 0,18:ATTRB 0,0:COLOR K,6:PRIN
T"JOUEUR ";GR$(K)
110 I=1:J=1
120 C=POINT(32*I+1,32*J+1)+1:C=ABS(C):IF
 C=7 THEN C=6
130 LOCATE 4*I,4*J,0:ATTRB 1,1:COLOR 0,6
:PRINT"X"
140 LOCATE 4*I,4*J,0:IF C<>6 THEN COLOR
6,C:PRINT GR$(C) ELSE COLOR 6:PRINT CHR$
(127)
150 A$=INKEY$
160 J=J-(A$=CHR$(10) AND J<3)+(A$=CHR$(1
1) AND J>1)
170 I=I-(A$=CHR$(9) AND I<3)+(A$=CHR$(8)
 AND I>1)
180 IF C<>6 OR A$<>"0" THEN 120
190 LOCATE 4*I,4*J:COLOR 6,K:PRINT GR$(K
)
200 K=ABS(K-1):COLOR K
210 IF K=1 THEN PLAY"DOREMI" ELSE PLAY"D
OMISO"
220 GOTO 100
```



Anagrammes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexe (2 joueurs).
Difficulté du jeu : selon le niveau.

LE JEU :
Des lettres sont choisies au hasard par le MO6, le nombre de lettres étant fonction du niveau de difficulté désiré. Puis les lettres du mot ainsi obtenu sont brouillées, et affichées au bas de l'écran ; un nouveau mot est alors confectionné, de façon aléatoire, à partir des mêmes lettres et il est affiché sous le premier, et ainsi de suite ; les deux joueurs (gauche et droit) disposent chacun d'une touche (1 ou Ø) et doivent la presser dès qu'ils voient s'afficher au bas de l'écran un mot déjà présent sur l'écran ; leur score est alors évalué, et un message correspondant à leur résultat sera délivré.

LE PROGRAMME :
Le niveau du jeu est entré à la ligne 5Ø. La boucle I (8Ø-1ØØ) réalise le choix des lettres.
Une deuxième boucle I (12Ø-19Ø) effectue le brouillage, et la ligne 14Ø permet de détecter la pression d'une touche.
La ligne 2ØØ affiche un nouveau mot, et fait monter d'une ligne sur l'écran les mots déjà obtenus.
La ligne 24Ø cherche quel est celui des deux joueurs qui a pressé une touche.
La boucle I (25Ø-32Ø) explore les mots présents sur l'écran ligne par ligne, alors que la boucle J (27Ø-3ØØ) saisit un par un les caractères de ces mots. On obtient ainsi un nouveau mot qui est comparé au mot initial en ligne 31Ø et le résultat BRAVO apparaît si c'est exact.
Sinon, la partie s'arrête sur l'affichage PERDU (ligne 33Ø).

POUR JOUER :
On lance le programme par RUN.
On devra ensuite entrer le niveau de difficulté (en principe de 1 à 3) et faire ENTREE. Les mots s'affichent en partant du bas de l'écran, et montent au fur et à mesure ; si l'un des joueurs croit reconnaître dans le mot qui apparaît un des mots déjà affichés, il devra immédiatement presser sa touche (1 ou Ø), et son résultat sera affiché après un court instant.
La partie sera alors terminée.

EXTENSIONS POSSIBLES :
On pourra utiliser d'autres affichages que des mots, modifier les temps d'affichage, faire participer plus de deux joueurs, améliorer le son et la couleur, etc.

```
10 REM ANAGRAMMES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 INPUT"NIVEAU ( DE 1 A 3 ) :";N
60 IF N<1 OR N>3 THEN 40
70 S$=""
80 FOR I=1 TO N+3
90 S$=S$+CHR$(INT(RND*26+65))
100 NEXT I
110 A$="":B$=S$
120 FOR I=1 TO LEN(S$)
130 N=INT(RND*LEN(S$)+1)
140 D$=INKEY$
150 IF D$="1" OR D$="0" THEN 240
160 IF MID$(B$,N,1)="-" THEN 130
170 A$=A$+MID$(B$,N,1)
180 B$=LEFT$(B$,N-1)+"-"+RIGHT$(B$,LEN(B
$)-N)
190 NEXT I
200 LOCATE 0,24:PRINT A$
210 FOR I=1 TO 20*LEN(B$):NEXT I
220 S$=A$
230 GOTO 110
240 IF D$="1" THEN H$="JOUEUR DE GAUCHE"
 ELSE H$="JOUEUR DE DROITE"
250 FOR I=0 TO 22
260 B$=""
270 FOR J=0 TO LEN(S$)-1
280 R$=CHR$(SCREEN(J,I))
290 B$=B$+R$
300 NEXT J
310 IF B$=S$ THEN LOCATE LEN(S$)+4,I:PRI
NT"BRAVO , ";H$:PLAY"DOMISO":END
320 NEXT I
330 LOCATE LEN(S$)+4,24:PRINT"PERDU ";H$
;:PLAY"DODODO"
```

Œufs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

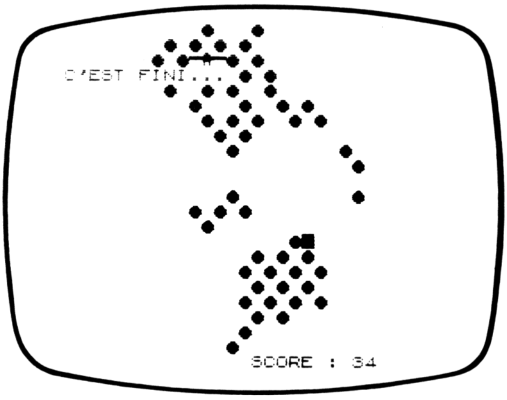
LE JEU :
Un oiseau mystérieux se déplace en battant des ailes au hasard sur l'écran. Il pond des œufs (représentés par des O) que vous (un carré de couleur) devez essayer de récupérer. Votre score (nombre d'œufs récupérés) sera affiché en permanence, et la partie se déroule en un laps de temps défini.

LE PROGRAMME :
L'oiseau est dessiné en deux parties (pour le mouvement) aux lignes 70 et 110 ; son déplacement aléatoire est géré par le sous-programme 210-300 (en particulier les lignes 260 à 290).
La ligne 130 permet de reboucler le programme sur la ligne 70.
Le déplacement du joueur est régi par les lignes 150 et 160, et le test permettant de détecter la présence d'un œuf est effectué en ligne 170.
La ligne 190 permet l'affichage du score.
La fin de jeu est détectée en ligne 230.

POUR JOUER :
On lancera le programme par RUN.
L'oiseau commence alors à se déplacer au hasard sur l'écran, et se met à pondre ses œufs. Lorsque le symbole qui représente le joueur (un carré de couleur) apparaît sur l'écran, celui-ci peut se déplacer en utilisant les 4 touches de déplacement. Il vous faut dès lors essayer de récupérer un maximum d'œufs avant la fin de la partie.
On pourra arrêter de jouer à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Sur le même principe, on pourra facilement imaginer d'autres thèmes de jeu. On pourra bien sûr améliorer la couleur, les sons, le graphisme, et éventuellement, définir un jeu à plusieurs joueurs, chacun essayant de récupérer un maximum d'œufs.

```
10 REM OEUFS
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,5:DEFGR$(0)=0,28,62,127,127,12
7,62,28:DEFGR$(1)=24,60,255,255,60,36,36
,0:DEFGR$(2)=0,0,1,3,6,12,24,48:DEFGR$(3
)=0,0,128,192,96,48,24,12:DEFGR$(4)=0,0,
255,255,0,0,0,0
50 CLS:P=200:R=18:S=10:T=0
60 X=INT(RND*36):Y=INT(RND*21)
70 LOCATE X,Y,0:COLOR 0,6:PRINT GR$(2)+G
R$(1)+GR$(3)
80 FOR I=1 TO 30 : A$=INKEY$
90 IF A$="" THEN NEXT I ELSE GOSUB 140
100 GOSUB 210
110 LOCATE X,Y,0:COLOR 0,6:PRINT GR$(4)+
GR$(1)+GR$(4)
120 GOSUB 210
130 GOTO 70
140 LOCATE R,S:COLOR 0,6:PRINT " "
150 R=R-(A$=CHR$(9) AND R<36) + (A$=CHR$
(8) AND R>0)
160 S=S-(A$=CHR$(10) AND S<21) + (A$=CHR
$(11) AND S>0)
170 G=POINT(8*R+4,8*S+4)
180 T=T - (G=1)
190 LOCATE 15,22:COLOR 2,0:PRINT"SCORE :
";T
200 RETURN
210 A=X:B=Y
220 P=P-1
230 IF P=0 THEN PLAY"DOREMI":PRINT"C'EST
 FINI...":END
240 LOCATE R,S:COLOR 3:PRINT CHR$(127)
250 LOCATE A,B:COLOR 1,6:PRINT " "+GR$(0
)+" "
260 X=ABS(X+INT(1+2*(RND<0.5)))
270 Y=ABS(Y+INT(1+2*(RND<0.5)))
280 IF X>36 THEN X=1
290 IF Y>21 THEN Y=1
300 RETURN
```



Taquin de lettres

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

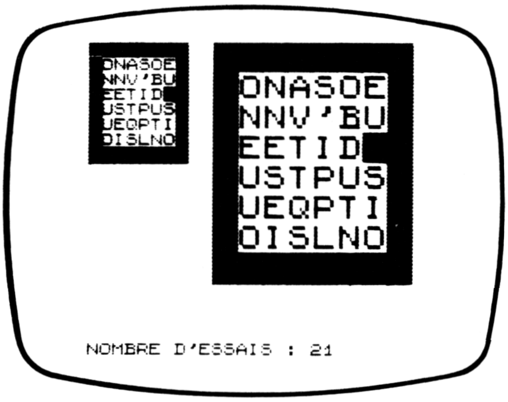
LE JEU :
Qui d'entre nous n'a pas, un jour ou l'autre, eu dans les mains un de ces petits puzzles où l'on essaie de déplacer d'une case à une autre des petits carrés de plastique colorés, jusqu'à l'obtention d'un résultat bien précis ? En son temps le 14-15 de SAM LOYD, a eu autant de succès que notre célèbre RUBIK'S.
Le taquin qui vous est proposé ici utilise 35 lettres, disposées au hasard dans un carré de 6 cases sur 6 cases, et ces lettres peuvent, replacées dans leur ordre initial, vous permettre de reconstituer un proverbe très connu. Une case coloriée vous permet de déplacer les lettres, et un compteur affiche en permanence le nombre d'essais : arriverez-vous à retrouver le proverbe ?

LE PROGRAMME :
La phrase à reconstituer (qui pourra bien sûr être modifiée) est placée dans une variable R$ en ligne 6Ø.
Le mélange des lettres et l'impression sont faits par la boucle I (7Ø-16Ø).
La ligne 19Ø réalise l'affichage du carré coloré ; celui-ci est déplacé par les lignes 23Ø et 24Ø en utilisant les 4 flèches.
Les lignes 25Ø et 26Ø réimpriment la lettre concernée à l'emplacement du carré coloré.
La ligne 28Ø comptabilise le nombre d'essais qui est imprimé grâce à la ligne 18Ø.

POUR JOUER :
On lance le programme par RUN.
La partie sera évidemment d'autant plus difficile que l'on ne connaît pas le proverbe à reconstituer. On peut déplacer les lettres et les faire passer à l'emplacement du carré coloré, à l'aide des 4 touches de direction (→, ←, ↓, ↑).
Le nombre d'essais est affiché en permanence, à droite du puzzle.
On peut arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
Le même principe peut être appliqué à toute sorte de taquins (manipulation de nombres, de couleurs, de sons éventuellement). On peut aussi prévoir plusieurs phrases, avec un choix effectué par un générateur aléatoire, etc.

```
10 REM TAQUIN DE LETTRES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 4,6:BOXF(0,0)-(64,64),0
50 BOXF(80,0)-(208,128),7
60 A$="":R$="ONASOUVENTBESOIND'UNPLUSPET
ITQUESOI"
70 FOR I=1 TO 35
80 X=INT(RND*35+1)
90 IF MID$(R$,X,1)="-" THEN 80
100 A$=A$+MID$(R$,X,1)
110 R$=LEFT$(R$,X-1)+"-"+RIGHT$(R$,LEN(R
$)-X)
120 F=ASC(MID$(A$,I,1))
130 X=I-6*INT(I/6):Y=INT(I/6)
140 LOCATE X+1,Y+1:COLOR 4,6:ATTRB 0,0:P
RINT CHR$(F)
150 LOCATE 2*X+12,2*Y+3:ATTRB 1,1:COLOR
1,3:PRINT CHR$(F)
160 NEXT I
170 X=0:Y=0:S=0:C=32
180 LOCATE 0,20:ATTRB 0,0:PRINT"NOMBRE D
'ESSAIS :";S
190 LOCATE X+1,Y+1:COLOR 1,6:ATTRB 0,0:P
RINT CHR$(127):LOCATE 2*X+12,2*Y+3,0:COL
OR 2,3:ATTRB 1,1:PRINT CHR$(127)
200 T$=INKEY$
210 IF T$="" THEN LOCATEX+1,Y+1:ATTRB 0,
0:COLOR 4,6:PRINT" ":LOCATE2*X+12,2*Y+3,
0:ATTRB 1,1:COLOR 1,3:PRINT" ":GOTO 190
220 I=X:J=Y
230 X=X - (T$=CHR$(9) AND X<5) + (T$=CHR
$(8) AND X>0)
240 Y=Y - (T$=CHR$(10) AND Y<5) + (T$=CH
R$(11) AND Y>0)
250 C=SCREEN(X+1,Y+1)
260 LOCATE I+1,J+1:COLOR 4,6:ATTRB 0,0:P
RINT CHR$(C):LOCATE 2*I+12,2*J+3:ATTRB 1
,1:COLOR 1,3:PRINT CHR$(C)
270 IF I=X AND J=Y THEN PLAY"DODODO":GOT
O 290
280 S=S+1:PLAY"DOMISO"
290 GOTO 180
```



Cibles

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : hasard.
Difficulté du jeu : facile.

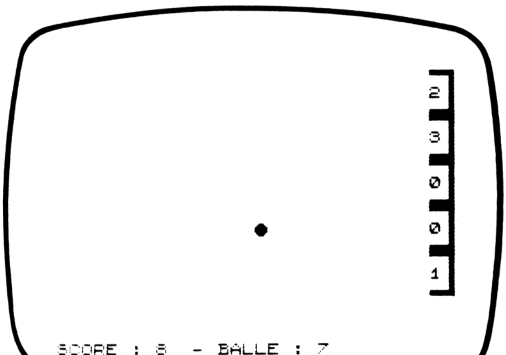
LE JEU :
Il consiste simplement à tirer sur des cibles, en essayant de marquer le maximum de points ; le nombre de cibles (de 1 à 6) sera choisi en début de partie, et le score réalisé sera affiché en permanence. Si l'on touche le bord d'une cible, la partie s'arrête. On marquera d'autant plus de points que le nombre sur lequel on tombe sera plus grand.

LE PROGRAMME :
Le nombre de cibles est entré en ligne 5Ø. La boucle I (8Ø-1ØØ) effectue l'impression des cibles.
La ligne 12Ø affichera le score en permanence.
La balle est affichée par la ligne 13Ø.
La boucle I (15Ø-2ØØ) réalise l'impression des chiffres contenus dans les cibles, et permet le déplacement de la boule (ligne 19Ø).
La ligne 22Ø regarde si l'on a tiré, et la ligne 23Ø regarde si la boule est encore sur la ligne de départ.
La ligne 24Ø fait progresser la boule d'une case vers la droite, et la ligne 25Ø vérifie si l'on a ou non atteint les cibles ; puis les lignes 26Ø et 27Ø regardent ce que l'on a touché.
La ligne 28Ø incrémente le score.
Les lignes 29Ø et 3ØØ s'occupent de la fin de partie.

POUR JOUER :
Lancer le programme par RUN, et taper le nombre de cibles désiré suivi de ENTREE. Les cibles s'affichent, et la partie commence. Votre balle, représentée par un O est affichée en bas à gauche ; vous pouvez la déplacer avec les flèches haut et bas (touches ↑ et ↓), et vous pouvez tirer à tout moment en tapant Ø.
Le score sera modifié en fonction de votre résultat, et une nouvelle balle apparaît. Il est possible d'arrêter à tout moment en tapant CNT et C, sinon le jeu s'arrête au bout d'un certain nombre de tirs.

EXTENSIONS POSSIBLES :
On pourra sans difficulté augmenter le nombre de cibles, modifier la règle d'apparition des nombres dans les cibles pour la rendre moins hasardeuse, s'arranger pour que la balle elle-même soit représentée par un chiffre qui devrait coïncider à l'arrivée avec celui de la cible.

```
10 REM CIBLES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,6:DEFGR$(0)=0,28,62,127,127,12
7,62,28:DEFGR$(1)=255,255,255,7,7,7,7,7:
DEFGR$(2)=7,7,7,7,7,7,7,7:DEFGR$(3)=7,7,
7,7,7,255,255,255:DEFGR$(4)=255,255,255,
0,0,0,0,0:DEFGR$(5)=0,0,0,0,0,255,255,25
5
50 PRINT:INPUT"NOMBRE DE CIBLES :";N
60 P=10:J=30:S=0:B=1
70 CLS
80 FOR I=1 TO N
90 LOCATE J,3*I-1:PRINT GR$(4);GR$(1):LO
CATE J,3*I:PRINT " ";GR$(2):LOCATE J,3*I
+1:PRINT GR$(5);GR$(3)
100 NEXT I
110 K=0:A=18
120 LOCATE 0,20:COLOR 4,6:PRINT"SCORE :"
;S;" - BALLE :";B
130 LOCATE K,A,0:COLOR 1,6:PRINT GR$(0)
140 D=A
150 FOR I=1 TO N
160 U=INT(RND*P)
170 LOCATE J,3*I,0:COLOR 0,7:PRINT CHR$(
U+48)
180 A$=INKEY$
190 A=A - (A$=CHR$(10)) + (A$=CHR$(11))
200 NEXT I
210 LOCATE K,D,0:COLOR 4,6:PRINT " "
220 IF A$="0" THEN B=B+1:GOTO 240
230 IF K=0 THEN 130
240 K=K+1
250 IF K<J THEN 130
260 C=SCREEN(K,A)
270 IF C<48 OR C>57 THEN PRINT"PERDU..."
:PLAY"SOMIDO":END
280 S=S+C-48:PLAY"DOMISO"
290 P=P-1
300 IF P=0 THEN PLAY"DOREMI":END
310 GOTO 110
```



Gouttes

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : dessin.
Difficulté du jeu : aucune.

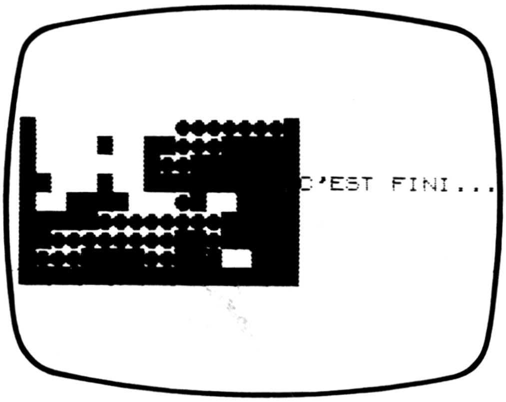
LE JEU :
C'est un véritable programme de dessin animé que l'on peut considérer, en fait, comme un interlude ; un récipient est dessiné sur l'écran, vu en coupe, et il contient déjà des obstacles. Une goutte d'eau, symbolisée par un rond gris, apparaît au-dessus du récipient et tombe en remplissant le récipient au fur et à mesure. Sa trajectoire, lorsqu'elle rencontre un obstacle, est entièrement déterminée par le hasard : elle part à droite ou à gauche, jusqu'à ce qu'il n'y ait plus aucune possibilité. Alors, une autre goutte tombe, et ainsi de suite, jusqu'au débordement final.

LE PROGRAMME :
Les deux boucles emboîtées I (6Ø-13Ø) et J (7Ø-12Ø) dessinent le récipient et les obstacles.
La ligne 17Ø dessine la goutte et la ligne 21Ø envoie dans un sous-programme (37Ø-39Ø) qui permet de déterminer s'il y a ou non rencontre avec un obstacle.
Si c'est non, on va en 29Ø, et la goutte continue de tomber.
Si c'est oui, la goutte part à droite ou à gauche (ligne 23Ø) selon la valeur définie pour K en ligne 15Ø.
On regarde de nouveau s'il y a ou non obstacle, ou si la goutte peut continuer à tomber (lignes 24Ø, 25Ø ou 26Ø).
Selon le résultat du test, on utilisera 27Ø, 29Ø, 31Ø ou 34Ø.

POUR JOUER :
Le RUN déclenche le dessin du récipient et des obstacles, la goutte apparaît, se met à tomber et finit par se trouver une place. Une autre goutte apparaît et l'on recommence.
Le programme s'arrête de lui-même lorsque le récipient déborde ; on peut l'arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut, bien entendu, ajouter la couleur et le son et aussi envisager un jeu dans lequel un petit personnage devrait échapper aux gouttes ou à la noyade.

```
10 REM GOUTTES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,1:DEFGR$(0)=0,28,62,127,127,12
7,62,28
50 CLS:SCREEN ,7
60 FOR I=12 TO 20
70 FOR J=1 TO 16
80 LOCATE 0,I:COLOR 0:PRINT CHR$(127):LO
CATE 17,I:PRINT CHR$(127)
90 LOCATE J,20:PRINT CHR$(127)
100 IF I<>12 AND I<>20 THEN LOCATE J,I:C
OLOR 1:PRINT CHR$(127)
110 IF I<>20 AND RND<0.7 THEN LOCATE INT
(RND*15+1),I:PRINT"  "
120 NEXT J
130 NEXT I
140 Y=INT(RND*7+6):X=6
150 K=-2*(RND<0.5)-1
160 I=0
170 LOCATE Y,X,0:COLOR 6,7:PRINT GR$(0)
180 FOR J=0 TO 100:NEXT J
190 LOCATE Y,X,0:PRINT " "
200 U=X+1:V=Y
210 GOSUB 370
220 IF J=0 THEN 290
230 U=X:V=Y+K
240 GOSUB 370
250 IF J=0 THEN 310
260 IF I=0 THEN 340
270 LOCATE Y,X,0:COLOR 6,7:PRINT GR$(0)
280 GOTO 140
290 X=X+1
300 GOTO 150
310 Y=Y+K
320 I=1
330 GOTO 170
340 K=-K
350 I=1
360 GOTO 230
370 C=SCREEN(V,U):BEEP
380 J=-(C<>32):IF U=24 THEN LOCATE 18,15
:PRINT"C,EST FINI...":PLAY"DOMISO":END
390 RETURN
```



Obstacles

REFERENCES :
Difficulté de programmation : assez difficile.
Catégorie de jeu : jeu de réflexe (pour un joueur).
Difficulté du jeu : moyenne.

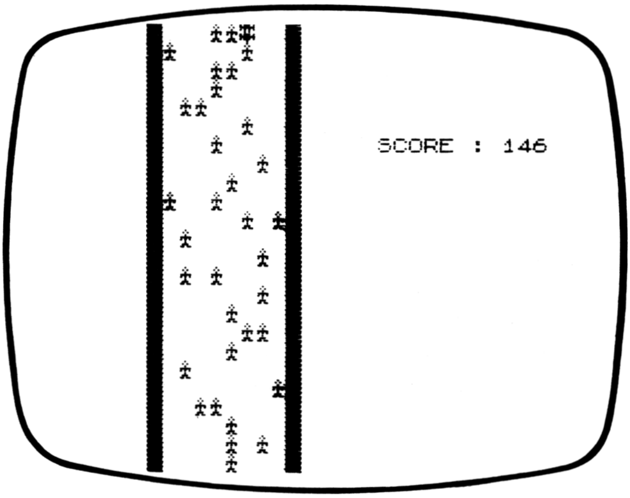
LE JEU :
Des piétons se déplacent sur l'écran pour venir à la rencontre d'une petite voiture qui dispose de la possibilité de se déplacer vers la gauche ou vers la droite pour les éviter et qui devra tenir le plus longtemps possible.
La distance parcourue sera affichée en permanence, et le jeu se termine lorsque la voiture rencontre, soit le bord du terrain, soit un piéton.

LE PROGRAMME :
La ligne 8Ø fait avancer les piétons vers la voiture du joueur qui est imprimée grâce à la ligne 7Ø.
Les lignes 8Ø, 9Ø, 1ØØ fabriquent la route (les lignes 9Ø et 1ØØ bien qu'identiques sont toutes deux indispensables, car la valeur de RND ne sera, en principe, pas la même sur ces deux lignes).
La ligne 12Ø affiche le score.
La ligne 14Ø permettra de déplacer le véhicule.
Les lignes 15Ø et 16Ø permettent de contrôler s'il y a ou non un obstacle et arrêtent le programme s'il y a lieu.
Le score est incrémenté en ligne 17Ø.

POUR JOUER :
On lance le programme en tapant RUN.
La route et les piétons apparaissent en bas de l'écran et montent vers la voiture. La voiture peut se déplacer à l'aide des flèches de déplacement horizontal (touches → et ←).
La rencontre avec un piéton ou la bordure interrompt le déroulement du programme et provoque l'affichage du score.
On peut, cependant, sortir du programme à tout moment en utilisant les touches CNT et C.

EXTENSIONS POSSIBLES :
Il sera assez facile d'améliorer le dessin de la voiture, de la route, des piétons. On pourra améliorer le son et la couleur, admettre un certain nombre d'accidents, enregistrer le meilleur score (pour plusieurs joueurs), éventuellement dessiner plusieurs voitures, etc.

```
10 REM OBSTACLES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLEAR,,2:DEFGR$(0)=93,127,85,20,93,12
7,93,8:DEFGR$(1)=8,20,8,62,42,8,28,54
50 CLS
60 S=0:Y=5
70 LOCATE Y,1:COLOR 0,6:PRINT GR$(0)
80 LOCATE 0,24:COLOR 1,6:PRINT CHR$(127)
+"        "+CHR$(127);
90 IF RND<0.7 THEN LOCATE INT(RND*8+1),2
4:COLOR 4,6:PRINT GR$(1);
100 IFRND <0.8 THEN LOCATE INT(RND*8+1),
24:COLOR 7,6:PRINT GR$(1)
110 LOCATE 15,5:COLOR 4,6:PRINT"        
       "
120 LOCATE 15,6:PRINT"SCORE :";S
130 A$=INKEY$
140 Y=Y - (A$=CHR$(9) AND Y<9) + (A$=CHR
$(8) AND Y>0)
150 C=POINT(8*Y+5,9)
160 IF C=4 OR C=7 OR C=1 THEN PLAY"DOREM
I":END
170 S=S+1
180 GOTO 70
```



Capture dans l'espace

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (1 seul joueur).
Difficulté du jeu : moyenne.

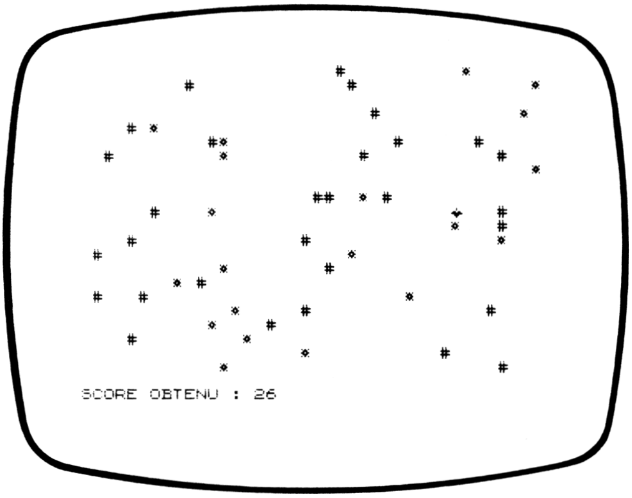
LE JEU :
Votre vaisseau spatial navigue dans un dangereux champ de météorites et votre mission est de vous emparer de la plus grande quantité possible de minerai d'urane, un métal indispensable que l'on ne peut trouver que dans ces lieux particulièrement dangereux.
Vous devez arriver à manœuvrer pour recueillir l'urane, sans vous faire heurter par une météorite.

LE PROGRAMME :
La ligne 7Ø permet de faire apparaître les météorites.
La ligne 9Ø fera apparaître le minerai d'urane et assure déplacement des météorites sur l'écran.
Les lignes11Ø à 13Ø contrôlent le déroulement du jeu (a-t-on heurté une météorite ou a-t-on capturé un bloc d'urane ?).
La rencontre d'une météorite provoque la fin de partie.
Votre vaisseau spatial est imprimé par les lignes 14Ø et 15Ø et son déplacement est contrôlé par la ligne 17Ø.

POUR JOUER :
On lance le programme par RUN et les météorites commencent à monter sur l'écran, ainsi que les blocs d'urane.
Les touches de déplacement horizontal (→ et ←) seront utilisées pour déplacer votre vaisseau et récupérer l'urane, mais le moindre heurt avec une météorite provoquera la fin de la partie et l'affichage de la quantité d'urane capturée.
On pourra aussi arrêter à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut rendre le jeu de plus en plus difficile en augmentant progressivement le nombre de météorites, ou encore, en minutant la durée de la partie.
On peut aussi prévoir un affichage permanent du score, introduire un deuxième vaisseau spatial manœuvré par une puissance étrangère et qui cherche aussi à s'emparer de l'urane...

```
10 REM CAPTURES DANS L'ESPACE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 4,6:CLEAR,,1:DEFGR$(0)=0,0
,8,8,54,127,28,8
50 X=15:Z=0
60 COLOR 4,6:S=INT(RND*40)
70 LOCATE S,23:COLOR 0,1:PRINT "#";
80 S=INT(RND*40)
90 IF RND<0.7 THEN LOCATE S,23:COLOR 0,3
:PRINT"*";:COLOR 4,6:PRINT:PRINT
100 FOR I=1 TO 10 : NEXT I
110 C=SCREEN(X,12)
120 IF C=42 THEN PRINT"SCORE OBTENU :";Z
:PLAY"DOMISO":END
130 IF C=35 THEN Z=Z+1:PLAY"DOREMI"
140 LOCATE X-1,11:COLOR 4,6:PRINT"    "
150 LOCATE X-1,12:COLOR 7,6:PRINT" "+GR$
(0)+" "
160 A$=INKEY$
170 X=X- (A$=CHR$(9) AND X<39) + (A$=CHR
$(8) AND X>0)
180 GOTO 60
```



Anomalie

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : solitaire.
Difficulté du jeu : moyenne.

LE JEU :
Il repose essentiellement sur une bonne capacité d'observation. Deux grilles de 25 cases chacune sont dessinées sur l'écran. Des symboles identiques sont affichés en des endroits différents des deux grilles, mais l'un des symboles a été modifié sur la grille de droite. Le jeu consiste à trouver lequel, en entrant ses coordonnées dans l'ordre ligne-colonne.

LE PROGRAMME :
La boucle I (5Ø-8Ø) imprime les chiffres de repérage des grilles.
La boucle principale I (1ØØ-16Ø) imprime les symboles des deux grilles, dans un ordre différent.
La ligne 11Ø choisit le caractère à imprimer.
Le sous-programme (1ØØØ-1Ø4Ø) s'occupe de faire l'impression à un emplacement libre (on l'appelle deux fois pour les deux grilles).
Les lignes 17Ø et 18Ø changent, de façon aléatoire, l'un des caractères.
Les coordonnées du point essayé sont entrées aux lignes 19Ø et 2ØØ, et le contrôle est effectué par les lignes 21Ø et 22Ø.
Les lignes 23Ø et 24Ø s'occupent de l'affichage du résultat.

POUR JOUER :
Après avoir lancé le programme par l'instruction RUN, on obtient l'affichage des deux grilles. On devra alors chercher, par comparaison, quel est le caractère qui a été modifié (sur la grille de droite).
On entre alors son numéro de ligne, suivi de ENTREE, et son numéro de colonne, suivi de ENTREE.
Le programme répondra alors BRAVO ou PERDU.

EXTENSIONS POSSIBLES :
Il sera possible d'utiliser des symboles graphiques de diverses couleurs, ce qui pourra simplifier la recherche.
On peut aussi envisager d'utiliser des dessins plus sophistiqués, améliorer le contrôle du résultat, admettre la bonne réponse sur la grille de droite ou celle de gauche, introduire un minutage, etc.

```
10 REM ANOMALIE
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 4,6:T=5
50 FOR I=1 TO 5
60 LOCATE 2*I+1,0:COLOR 7,0:PRINT I:LOCA
TE 0,2*I:PRINT I
70 LOCATE 2*I+19,0:COLOR 1,0:PRINT I:LOC
ATE 18,2*I:PRINT I
80 NEXT I
90 COLOR 6,6
100 FOR I=1 TO 22
110 D=INT(26*RND+1)
120 L=2:S=T
130 GOSUB 1000
140 L=11
150 GOSUB 1000
160 NEXT I
170 S=0
180 GOSUB 1000
190 SCREEN 0,2:LOCATE 0,16:INPUT"LIGNE..
...:";I
200 PRINT:INPUT"COLONNE...:";T
210 U=2*I:V=2*T+20
220 C=SCREEN(V,U)
230 IF D+64=C THEN PRINT "BRAVO...":PLAY
"DOMISO" ELSE PRINT"PERDU":PLAY"DODODO"
240 PRINT:PRINT"LA LETTRE A TROUVER ETAI
T :";CHR$(D+64):END
1000 U=2*INT(RND*T+1):V=2*INT(RND*S+L)
1010 C=SCREEN(V,U)
1020 IF C<>32 AND S<>0 THEN 1000
1030 LOCATE V,U,0:PRINT CHR$(D+64)
1040 RETURN
```

```
     1 2 3 4 5          1 2 3 4 5

  1  B N X V B       1  O G H U

  2  W S V H         2  U X D V W

  3  D T U U W       3  V N   U O

  4  N   G U O       4  N S B W N

  5  V Y   N N       5  V N T B



LIGNE.....:? 4

COLONNE...:? 2
PERDU

LA LETTRE A TROUVER ETAIT :O
```

Plusieurs en un

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : variable, selon l'utilisation.

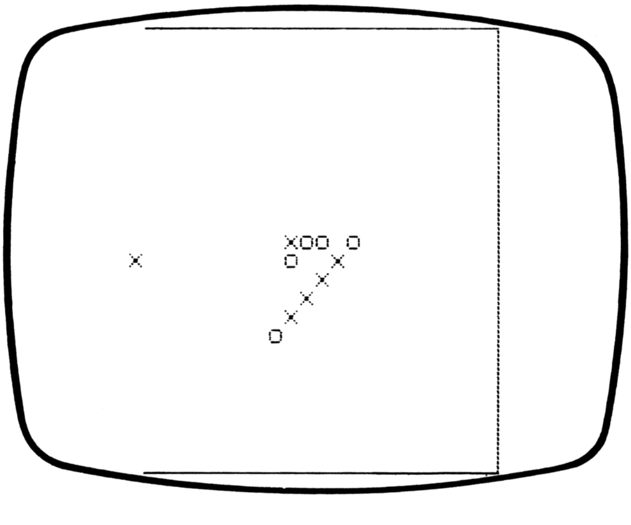
LE JEU :
Plusieurs jeux sont en réalité possibles avec ce programme qui va se comporter comme un crayon électronique vous permettant d'inscrire des symboles O ou X, en n'importe quel emplacement de votre feuille-écran de télévision. On pourra, par exemple, l'utiliser pour un jeu de morpion (le premier à réaliser un alignement de 5 pions identiques gagne) ou pour un GO (on réalise des encadrements des pions adverses) ou tout autre jeu, selon son imagination.

LE PROGRAMME :
Le terrain de jeu est délimité par un cadre mis en place par la ligne 3Ø.
Le changement de joueur est effectué par la ligne 5Ø.
Le joueur qui a le trait voit le symbole qu'il a choisi (O ou X) se placer en limite du cadre (ligne 7Ø).
Les lignes 1ØØ et 11Ø permettent aux deux joueurs de déplacer le symbole sur le terrain.
Les lignes 13Ø et 14Ø contrôlent si le déplacement est possible (case libre et non dépassement des limites).
La ligne 19Ø permet d'écrire sur une case libre.

POUR JOUER :
RUN provoque l'affichage des limites du terrain et du numéro du joueur, ainsi que des symboles correspondants (O ou X) que l'on devra déplacer sur l'écran à l'aide des 4 touches de direction (→, ←, ↓ , ↑).
Il n'est pas possible de venir sur une case déjà occupée et il faudra appuyer sur la touche Ø pour inscrire son propre symbole dans une case disponible.
Dès que cela a été fait, le symbole du second joueur apparaît dans la colonne de gauche, à peu près au milieu, et le joueur procédera de la même manière que son adversaire.
Pour sortir du programme, il suffira de taper CNT et C.

EXTENSIONS POSSIBLES :
On pourra utiliser l'écran tout entier comme terrain de jeu et réaliser des contrôles en fonction des différents jeux que l'on désire pratiquer : contrôle d'alignements de 5 pièces pour le morpion, contrôle du nombre de pièces capturées pour le GO.
On pourra utiliser le crayon optique pour désigner la case à occuper.

```
10 REM PLUSIEURS EN UN
20 CLS:SCREEN 4,6
30 BOX(8,0)-(190,190),0
40 J=0
50 J=ABS(J-1)
60 X=12:Y=0
70 LOCATE Y,X,0:COLOR 1-3*(J=1),7:PRINT
CHR$(79-9*(J=1))
80 A$=INKEY$:IF A$="" THEN 80
90 A=X:B=Y
100 X=X-(A$=CHR$(10) AND X<23)+(A$=CHR$(
11) AND X>0)
110 Y=Y-(A$=CHR$(9) AND Y<23)+(A$=CHR$(8
) AND Y>1)
120 LOCATE B,A,0:COLOR 4,6:PRINT" "
130 C=SCREEN(Y,X)
140 IF C<>79 AND C<>88 THEN 160
150 X=A:Y=B
160 FOR I=1 TO 10:A$=INKEY$
170 IF A$="" THEN NEXT I
180 IF A$<>"0" OR Y=0 THEN 70
190 LOCATE Y,X,0:COLOR 1-3*(J=1),6:PRINT
 CHR$(79-9*(J=1))
200 IF J=0 THEN PLAY"DO" ELSE PLAY"SO"
210 GOTO 50
```



Parcours du cavalier

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion.
Difficulté du jeu : difficile.

LE JEU :
Il se présente, ici, comme un jeu de hasard total, mais provient, en réalité, d'un problème assez difficile à résoudre : comment déplacer un cavalier sur un échiquier de façon qu'il passe une fois, et une seule, sur chacune des 64 cases du damier.
Le cavalier se déplace suivant les règles communément admises aux échecs, et sa position de départ est définie par le joueur ; par contre, ses déplacements sont effectués au hasard. Arrivera-t-il à parcourir les 64 cases ?

LE PROGRAMME :
La ligne 60 définit les divers déplacements possibles, dans deux variables alphanumériques A$ et B$; la boucle emboîtée I (70-100) dessine le terrain de jeu.
On entre les coordonnées de la case de départ (ligne, puis colonne) aux lignes 110 et 120.
La ligne 140 imprime le numéro du coup.
La ligne 150 permet le tirage au hasard d'un déplacement du cavalier.
La boucle Z (160-210) cherche si le déplacement est possible ; si ce n'est pas le cas, on essaie tous les autres (toujours dans le même ordre).
Les lignes 170 et 180 effectuent le déplacement.
La ligne 190 contrôle que l'on ne sorte pas du terrain de jeu.
La ligne 200 envoie en ligne 230 si le coup est possible.
La ligne 220 gère la fin de partie (cavalier bloqué) et affiche le score.
Les lignes 230, 240 et 250 réalisent le coup et incrémentent le score.

POUR JOUER :
On lance le programme par l'ordre RUN. Le terrain de jeu s'affiche ; on choisit alors la position de départ en tapant d'abord un numéro de ligne (entre 1 et 8), suivi de ENTREE puis son numéro de colonne, suivi de ENTREE. Le jeu commence immédiatement et il se déroule de façon entièrement automatique jusqu'à ce que le cavalier soit bloqué. On peut, bien sûr, arrêter à tout moment par CNT et C.

EXTENSIONS POSSIBLES :
On peut imaginer de nombreuses variantes sur ce thème, ne serait-ce qu'en commençant par faire varier la taille des échiquiers, on peut aussi améliorer considérablement le graphisme. Mais, le plus intéressant est d'arriver à définir un algorithme donnant la ou les solutions du problème initial...

```
10 REM PARCOURS DU CAVALIER
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS
50 S=1:T=0
60 A$="56653223":B$="653223"+A$+"56":A$=
A$+A$
70 FOR I=1 TO 8
80 LOCATE 3*I+1,0:PRINT I
90 LOCATE 0,2*I:PRINT I;"  .  .  .  .  .
  .  .  ."
100 NEXT I
110 LOCATE 0,18:INPUT"LIGNE DE DEPART :"
;I
120 PRINT:INPUT"COLONNE DE DEPART :";J
130 IF S>9 THEN S=0:T=T+1
140 LOCATE 3*J+2,2*I,0:COLOR 1,7:PRINT R
IGHT$(STR$(S),1):IF T<>0 THEN LOCATE 3*J
+1,2*I,0:PRINT RIGHT$(STR$(T),1)
150 Z=INT(RND*8+1)
160 FOR Z=Z TO Z+8
170 A=I+VAL(MID$(A$,Z,1))-4
180 B=J+VAL(MID$(B$,Z,1))-4
190 IF A<1 OR B<1 OR A>8 OR B>8 THEN 210
200 IF SCREEN(3*B+2,2*A)=46 THEN 230
210 NEXT Z
220 LOCATE 25,18:PRINT"TOTAL :";T*10+S:P
LAY"DOMISO":END
230 I=A:J=B
240 S=S+1:PLAY"DOSO"
250 GOTO 130
```

```
     1  2  3  4  5  6  7  8
1    .  7 32  5 18 35 20 45
2   31  4 17  . 33 44 37  .
3    8  1  6 43 36 19 34 21
4    3 30  9 16 23  .  . 38
5   10 15  2 25 42  . 22  .
6   29 26 11  .  . 24 39  .
7   14  .  . 27 12 41  .  .
8    . 28 13  .  .  .  . 40
LIGNE DE DEPART :? 3      TOTAL : 45
COLONNE DE DEPART :? 2
```

Mur de briques

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexe (pour 1 joueur).
Difficulté du jeu : difficile.

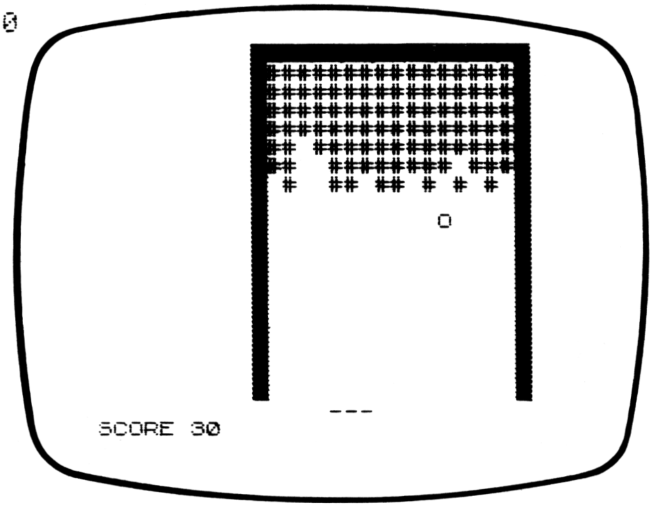
LE JEU :
S'il est un jeu que l'on peut présenter comme un grand classique des jeux d'arcades pour micro-ordinateur, c'est bien le mur de briques (en anglais BREAKOUT).
Il est offert, ici, dans une version simplifiée, mais offre néanmoins de bonnes performances : affichage du score en fin de partie, rebonds aléatoires, etc.
La règle en est très élémentaire : vous disposez d'une raquette qui peut être déplacée horizontalement pour renvoyer une balle (matérialisée par un point) qui va démolir les briques dessinées en haut de l'écran.
Votre problème est d'en démolir le plus grand nombre possible, sans rater la balle, ce qui provoquerait la fin de partie et l'affichage du score.

LE PROGRAMME :
La boucle I (5Ø-8Ø) dessine le terrain de jeu et les briques sont mises en place par les boucles emboîtées I (9Ø-13Ø) et J (1ØØ-12Ø).
La ligne 15Ø dessine la raquette dont la position est contrôlée par la ligne 17Ø.
Les lignes 22Ø à 25Ø regardent si l'on a touché le mur, les briques ou la raquette.
Les rebonds sont gérés en conséquence avec des éventuels aléas, par les lignes 26Ø et 3ØØ.
La ligne 19Ø affiche le score en permanence.

POUR JOUER :
Le programme sera lancé par l'instruction RUN, ce qui provoque l'affichage du terrain de jeu puis, très rapidement, l'envoi de la balle. Le joueur devra alors essayer de déplacer sa raquette à l'aide des touches de déplacement horizontal (→ et ←).
Si, à un moment quelconque, sa balle ne touche pas la raquette, le score sera affiché et la partie s'arrête.
On peut aussi terminer à tout moment par CNT et C.

EXTENSIONS POSSIBLES :
On peut envisager d'agrandir le terrain de jeu, d'augmenter le nombre de briques, de prévoir plusieurs balles, d'introduire des effets sonores.
On peut aussi, en s'inspirant des jeux d'arcades, introduire de petits ouvriers qui viendront reconstruire votre mur de briques, au fur et à mesure que vous le démolissez.

```
10 REM MUR DE BRIQUES
20 INPUT"DONNEZ UN ENTIER :";F
30 FOR I=1 TO F:Y=RND:NEXT I
40 CLS:SCREEN 0,7:A$=CHR$(127)
50 FOR I=1 TO 18
60 LOCATE 9+I,0:PRINT CHR$(61)
70 LOCATE 10,I:PRINT A$:LOCATE 27,I:PRIN
T A$
80 NEXT I
90 FOR I=1 TO 7
100 FOR J=11 TO 26
110 LOCATE J,I:COLOR 0,I:PRINT "#"
120 NEXT J
130 NEXT I
140 H=-1:K=-0.5:J=18:I=17:Y=J
150 LOCATE Y-2,19:COLOR 0,7:PRINT" --- "
160 A$=INKEY$
170 Y=Y - (A$=CHR$(9) AND Y<28) + (A$=CH
R$(8) AND Y>8)
180 IF T=1 THEN LOCATE J,I:PRINT " "
190 LOCATE 0,20:PRINT"SCORE"S
200 IF I>22 THEN PLAY"DODODO":END
210 I=I+H:J=J+K
220 C=SCREEN(J,I)
230 IF C=32 THEN LOCATE J,I:PRINT "O":T=
1:GOTO 150
240 IF C=127 THEN 280
250 IF C=35 THEN F=POINT(8*J+1,8*I+1):LO
CATE J,I:PRINT"O":T=1:GOTO 300
260 H=-H
270 K=RND-RND
280 K=-K:T=0
290 GOTO 150
300 S=S+9+F
310 K=-(K-(2*(RND<0.5)+1)*RND/3)
320 H=-H
330 GOTO 150
```



Othello

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (2 joueurs).
Difficulté du jeu : difficile.

LE JEU :
Devenu un grand classique des jeux sur micro-ordinateur, le jeu d'Othello ou Reversi gagne, même si comme ici il est pratiqué à deux joueurs, à être géré sur ordinateur, avec visualisation : en effet, le jeu consistant essentiellement en des retournements des pions de l'adversaire, ceci peut se révéler, à la longue, très fastidieux, surtout si vous êtes un joueur assidu.
Le principe est que le joueur possédant les X doit pour pouvoir jouer, réussir à encadrer une ou plusieurs des pièces de son adversaire, dans une ou plusieurs directions ; s'il y réussit, les O seront alors retournés pour devenir des X (le retournement pourra être fait horizontalement, verticalement ou suivant les diagonales).
Le joueur ayant les O procédera de la même façon, et ceci jusqu'à ce que le damier soit rempli ou que la personne ne puisse plus jouer (si un joueur ne peut retourner des pions, il doit passer son tour).

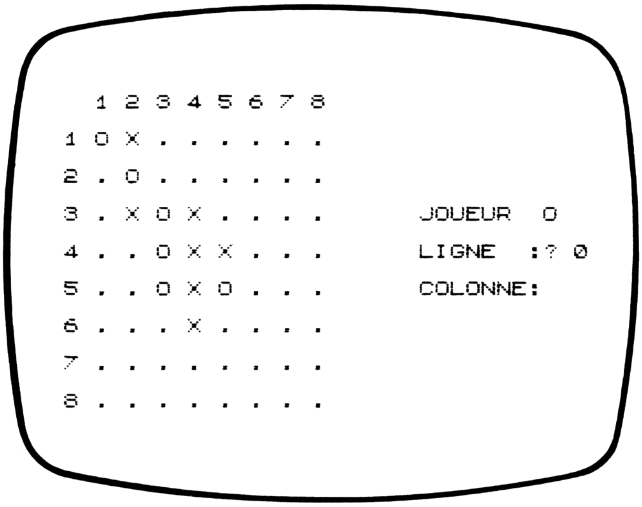
LE PROGRAMME :
Les boucles emboîtées P (5Ø-12Ø) et Q (7Ø-11Ø) dessinent le terrain de jeu et la ligne 14Ø affiche un X ou un O, selon le joueur qui commence.
La boucle P (2ØØ-32Ø) contrôle la validité du coup (la case est-elle libre : lignes 3ØØ et 31Ø), en utilisant une partie de programme qui se trouve aux lignes 36Ø à 42Ø (a-t-on pris un ou plusieurs ennemis en ligne 36Ø et retournement éventuel autorisé par la ligne 38Ø : on refait un balayage dans toutes les directions, si l'encadrement s'est révélé possible).
La ligne 2ØØ définit les 8 directions d'exploration.
Les lignes 24Ø et 25Ø calculent les déplacements à effectuer pour explorer ces 8 directions.
La ligne 23Ø réalise les déplacements.
La ligne 34Ø assure le changement de joueur, s'il y a eu retournement (contrôlé en ligne 33Ø).

POUR JOUER :
Après RUN, chaque joueur entrera à son tour les coordonnées (ligne, ENTREE, puis colonne, ENTREE) de la case où il désire jouer ; si le coup est accepté, les pions sont retournés et l'on change de joueur ; si un coup n'est pas possible, on entre Ø en première coordonnée pour le changement de joueur.

EXTENSIONS POSSIBLES :
Avant tout, bien sûr, améliorer l'affichage avec des couleurs, une grille plus aérée.
On pourra aussi imaginer un jeu machine pour le MO5.
Après avoir lancé le programme par RUN, il convient d'entrer la dimension de la grille (4, ENTREE ou 6, 8 ...).

```
10 REM OTHELLO
20 INPUT"DIMENSION DE LA GRILLE :";A
30 IF A/2<>INT(A/2) THEN GOTO 20
40 CLS:SCREEN 0,7
50 FOR P=1 TO A
60 LOCATE 2*P,0:PRINT RIGHT$(STR$(P),1)
70 FOR Q=1 TO A
80 LOCATE 0,2*P:COLOR 0,7:PRINT RIGHT$(S
TR$(P),1):LOCATE 2*Q,2*P:PRINT "."
90 IF Q=A/2+1 AND P=A/2 THEN LOCATE 2*Q-
2,2*P:COLOR 1,7:PRINT "O ";:COLOR 4,7:PR
INT "X"
100 IF Q=A/2+1 AND P=A/2+1 THEN LOCATE 2
*Q-2,2*P:COLOR 4,7:PRINT"X ";:COLOR 1,7:
PRINT"O"
110 NEXT Q
120 NEXT P
130 K=1
140 LOCATE 23,6,0:COLOR 0,1-3*(K=1):PRIN
T"JOUEUR ";:COLOR 1-3*(K=1),7:IF K=1 THE
N PRINT " X":PLAY"DOMISO" ELSE PRINT " O
":PLAY"DOREMI"
150 LOCATE 31,8:COLOR 0,7:PRINT"      ":L
OCATE31,10:PRINT"      "
160 LOCATE 23,8:INPUT"LIGNE  :";U
170 IF U=0 THEN 340
180 LOCATE 23,10:INPUT"COLONNE:";V
190 H=1
200 FOR P=1 TO 8
210 S=0
220 W=0
230 T=2*U:R=2*V
240 F=(P>3 AND P<7) - (P<3 OR P>7)
250 G=(P>1 AND P<5) - (P>=6)
260 Q=46 - 42*(S=1) - 33*(S=2)
270 IF S=0 THEN 290
280 LOCATE R,T,0:COLOR -1*(K=0)-4*(K=1),
7:PRINT CHR$(Q)
290 T=T+2*F:R=R+2*G
300 C=SCREEN(R,T)
310 IF C=79 OR C=88 THEN 360
320 NEXT P
330 IF H<>0 THEN 140
340 K=ABS(K-1)
350 GOTO 140
360 IF (C=79 AND K=1) OR (C=88 AND K=0)
```

```
THEN W=1
370 IF S<>0 THEN LOCATE R,T,0:COLOR -1*(
K=0)-4*(K=1):PRINT CHR$(Q)
380 IF W=1 AND ((C=79 AND K=1) OR (C=88
AND K=0)) THEN 290
390 IF W=0 OR S<>0 THEN 320
400 S=1 - (K=0)
410 H=-H*(S=0)
420 GOTO 220
```



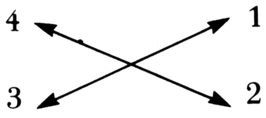
Dames

REFERENCES :
Difficulté de programmation : très difficile.
Catégorie de jeu : deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
C'est le jeu classique de dames à la française, mais un peu incomplet puisqu'il ne prévoit pas la possibilité de faire des dames. La règle du jeu reste classique, mais les contrôles ont été un peu simplifiés. Les mouvements des pièces sont entrés en une seule manœuvre (position de départ et direction du déplacement) et les prises sont réalisées automatiquement, si elles sont possibles.
Par contre le changement de joueur étant réalisé à chaque coup, il faudra éventuellement taper Ø pour pouvoir rejouer s'il y avait eu saut ou si le coup n'était pas possible.

LE PROGRAMME :
Les boucles I (8Ø-17Ø) et J (9Ø-16Ø) dessinent le damier et définissent les positions initiales des pions noirs et des pions blancs. Les positions sont dans une table R (X, Y) et la couleur dans une autre table C (X, Y).
L'impression des pions proprement dite est effectuée par les boucles emboîtées J (19Ø-25Ø) et I (2ØØ-24Ø).
La ligne 27Ø indique le joueur qui a le trait.
Le coup est entré en ligne 27Ø.
La ligne 28Ø réalise le changement de joueur (on tape Ø).
La ligne 3ØØ regarde si le déplacement est possible.
La ligne 33Ø traite le cas d'un déplacement et la ligne 34Ø celui d'un saut.
La ligne 38Ø assure le changement de joueur.

POUR JOUER :
Dès le lancement du programme par RUN, le damier est affiché avec ses pions, et un caractère O de couleur apparaît en dessous pour indiquer que le trait appartient au joueur correspondant.
On joue en indiquant par un mot de 3 lettres (pouvant commencer par un Ø) la ligne et la colonne de la pièce à déplacer, puis la direction du déplacement (codée de 1 à 4) suivant la convention :

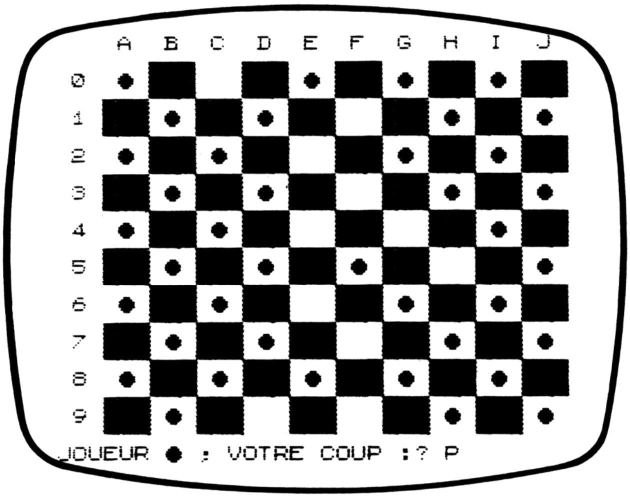


Par exemple 9D1 signifie que le O situé en ligne 9, colonne D, se déplace dans la direction 1 (ne pas oublier de faire ENTREE).
De même ØG3 signifiera pièce noire, ligne Ø, colonne G, direction 3.
Le changement de joueur étant effectué automatiquement il faudra taper Ø pour rejouer si le coup est refusé ou s'il y a eu prise (toujours terminer par ENTREE). Le programme ne contrôle pas l'arrivée à dame, et n'interdit pas les déplacements en arrière.

EXTENSIONS POSSIBLES :
Il sera assez facile de commencer par améliorer les contrôles (changement de joueur, coup en arrière qui ne correspondrait pas à une prise, sortie de terrain). On pourra envisager la création de dames, leur déplacement, et éventuellement écrire un programme permettant au MO6 de se mêler avec de bonnes chances de réussite à vos passionnantes parties.

```
10 REM DAMES
20 DIM R(10,10):DIM C(10,10):CLEAR,,4:DE
FGR$(0)=0,0,0,0,0,0,0,0:DEFGR$(1)=0,28,6
2,127,127,127,62,28:DEFGR$(2)=0,0,0,0,25
5,255,255,255:DEFGR$(3)=255,255,255,255,
0,0,0,0
30 A$=GR$(2)+GR$(2)+GR$(2)
40 B$=CHR$(127)+CHR$(127)+CHR$(127)
50 C$=GR$(3)+GR$(3)+GR$(3)
60 CLS:SCREEN 7,0
70 PRINT"    A  B  C  D  E  F  G  H  I
J"
80 FOR I=1 TO 10
90 FOR J=1 TO 10
100 IF (I+J)/2=INT((I+J)/2) THEN 120
110 LOCATE 3*J,2*I-1:PRINT A$:PRINT I-1:
LOCATE 3*J,2*I:PRINT B$:LOCATE 3*J,2*I+1
:PRINT C$
120 R(I,J)=0
130 IF (I+J)/2<>INT((I+J)/2) THEN 160
140 IF J<5 THEN R(I,J)=1 :C(I,J)=4
150 IF J>6 THEN R(I,J)=1:C(I,J)=1
160 NEXT J
170 NEXT I
180 H=1
190 FOR J=1 TO 10
200 FOR I=1 TO 10
210 IF (I+J)/2<>INT((I+J)/2) THEN 240
220 X=3*J+1:Y=2*I
230 LOCATE X,Y,0:COLOR C(I,J):PRINT GR$(
R(I,J))
240 NEXT I
250 NEXT J
260 LOCATE 25,22:PRINT"         ":PLAY"DO;RE
;MI"
270 LOCATE 0,22:COLOR 1-3*(H=1):PRINT"JO
UEUR ";GR$(1);:INPUT" ; VOTRE COUP :";D$
280 IF D$="0" THEN H=ABS(H-1):GOTO 260
```

```
290 GOSUB 400
300 IF R(L,C)=0 OR (C(L,C)=1 AND H=1) OR
 (C(L,C)=4 AND H=0) THEN 260
310 T=(1 AND (D=1 OR D=2)) - (1 AND (D=4
 OR D=3))
320 S=(1 AND (D=3 OR D=2)) - (1 AND (D=1
 OR D=4))
330 IF R(L+S,C+T)=0 THEN R(L,C)=0:R(L+S,
C+T)=1:C(L+S,C+T)=1-3*(H=1):GOTO 380
340 IF C(L+S,C+T)=1-3*(H=0) AND R(L+S,C+
T)=1 AND R(L+2*S,C+2*T)=0 THEN 360
350 GOTO 260
360 R(L,C)=0:R(L+S,C+T)=0:R(L+2*S,C+2*T)
=1:C(L+2*S,C+2*T)=1-3*(H=1)
370 GOTO 190
380 H=ABS(H-1)
390 GOTO 190
400 L=VAL(LEFT$(D$,1))+1:C=ASC(MID$(D$,2
,1))-64:D=VAL(RIGHT$(D$,1))
410 RETURN
```



Horloge 2

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : simulation.
Difficulté du jeu : aucune.

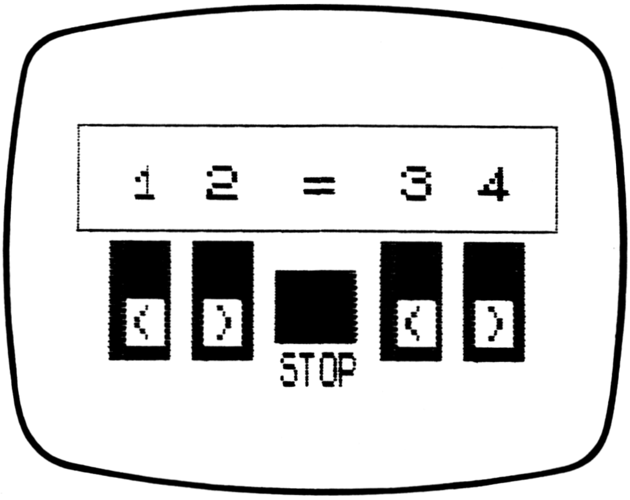
LE JEU :
Ce programme simule une horloge digitale grand format en utilisant les caractères d'origine du MO6, notablement agrandis. Il présente la particularité de faire appel au crayon optique du MO6 pour la mise à l'heure, et éventuellement l'arrêt ou le départ de l'horloge.

LE PROGRAMME :
Les lignes 3Ø à 9Ø dessinent les cinq zones destinées à l'utilisation du crayon optique, et représentent les symboles associés (défilement en avant par <, en arrière par > et arrêt : STOP).
La ligne 1ØØ sépare les heures et les minutes. Les heures sont affichées par la ligne 11Ø, et les minutes par la ligne 12Ø. Les boucles J (15Ø-22Ø) et I (16Ø-18Ø) effectuent l'affichage clignotant des secondes.
La saisie des informations en provenance du crayon optique est faite aux lignes 19Ø et 195.
Les informations sont traitées par les lignes 29Ø, 31Ø, 33Ø, 35Ø et 37Ø.
La ligne 25Ø contrôle le passage à 6Ø minutes.
La ligne 27Ø contrôle le passage à 24 heures.

POUR JOUER :
Faire RUN, l'heure est immédiatement affichée sous la forme ØØ = ØØ ; les secondes clignotent au milieu de l'écran.
Deux symboles < et >, sous les heures, permettent la mise à l'heure : il suffit de pointer la zone blanche correspondante avec le crayon optique pour faire défiler les heures en avant ou en arrière. De même pour les minutes.
On touchera la zone STOP pour arrêter l'horloge dans l'attente du prochain top sonore. La même zone STOP fera repartir l'horloge.
Faire CNT et C pour arrêter le programme. On pourra régler l'horloge en jouant sur la valeur de I, en ligne 16Ø.

EXTENSIONS POSSIBLES :
On peut améliorer l'affichage des secondes, ajouter une possibilité de réveil, prévoir un agenda automatique, faire apparaître un coucou chantant au moment du passage de l'heure.

```
10 REM HORLOGE
20 TI$="0000"
30 CLS:SCREEN 0,2
40 BOXF(74,100)-(98,140),7:BOXF(108,100)
-(132,140),7
50 BOXF(186,100)-(210,140),7:BOXF(220,10
0)-(244,140),7
60 BOX(60,60)-(258,96),0:BOXF(142,110)-(
174,134),7
70 LOCATE 18,18:ATTRB 0,1:PRINT"STOP"
80 LOCATE 10,16:COLOR 0,1:ATTRB 1,1:PRIN
T"<":LOCATE 14,16:PRINT">"
90 LOCATE 24,16:PRINT"<":LOCATE 28,16:PR
INT">"
100 H$=LEFT$(TI$,2):M$=RIGHT$(TI$,2)
110 LOCATE 10,10:COLOR 1,2:ATTRB 1,1 :PR
INT LEFT$(H$,1);TAB(13):PRINT RIGHT$(H$,
1)
120 LOCATE 24,10:COLOR 4,2:PRINT LEFT$(M
$,1);TAB(27):PRINT RIGHT$(M$,1)
130 M=VAL(M$):H=VAL(H$)
140 F=1
150 FOR J=1 TO 118
160 FOR I=1 TO 20
170 LOCATE 19,10,0:IF F=1 THEN COLOR 4,6
:PRINT"=" ELSE COLOR 0,2:PRINT" "
180 NEXT I
190 IF PTRIG=0 THEN 210
195 INPEN X,Y:S=(Y>100)*(Y<120)*((X>80)*
(X<100)+2*(X>112)*(X<132)+3*(X>190)*(X<2
10)+4*(X>224)*(X<244)+5*(X>144)*(X<176))
200 ON S GOTO 290,310,330,350,370
210 F=ABS(F-1)
220 NEXT J
230 M=M+1
240 M$=STR$(M):M$="0"+RIGHT$(M$,LEN(M$)-
1):M$=RIGHT$(M$,2)
250 IF M=60 THEN M$="00":H=H+1
260 H$=STR$(H):H$="0"+RIGHT$(H$,LEN(H$)-
1):H$=RIGHT$(H$,2)
270 IF H=24 THEN H$="00"
280 GOTO 110
290 H=H+1:IF H>23 THEN H=0
300 GOTO 360
310 H=H-1:IF H<0 THEN H=23
320 GOTO 360
330 M=M+1:IF M>59 THEN M=0
340 GOTO 360
350 M=M-1:IF M<0 THEN M=59
360 GOTO 240
370 IF PTRIG=-1 THEN 370
380 GOTO 240
```


1 2 = 3 4
<
>
<
>
STOP

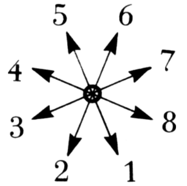
Quatre à quatre

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : réflexion (un ou deux joueurs).
Difficulté du jeu : difficile.

LE JEU :
Il pourra être pratiqué à un ou plusieurs joueurs. Dans les deux cas, le but est d'arriver à échanger les pions X avec les pions O ; si l'on joue en solitaire, cet échange devra être réalisé avec le moins de coups possible. Les pions sont déplacés alternativement en respectant les règles de déplacement du cavalier aux échecs ; il n'y a pas de prise, et on ne peut passer son tour.

LE PROGRAMME :
La variable alphanumérique A$, en ligne 4Ø, contient la table des coups possibles (déplacement du cavalier).
La boucle I (6Ø-1ØØ) dessine le terrain de jeu, et les positions de départ.
La ligne 12Ø permet d'afficher O ou X, selon le joueur qui a le trait.
La ligne 15Ø permet l'affichage du curseur clignotant (ici un pavé de couleur) qui indique sur quelle case on se trouve.
Les lignes 18Ø et 19Ø permettent le déplacement du curseur.
La ligne 2ØØ regarde si l'on a appuyé sur une touche de 1 à 8, donnant la direction du déplacement.
La ligne 21Ø réimprime la figure qui était sous le curseur. La ligne 24Ø transforme en nombre le symbole chiffre frappé. La ligne 23Ø réalise le contrôle de départ.
Les lignes 25Ø et 26Ø effectuent le déplacement. La ligne 29Ø réalise le contrôle d'arrivée. Le résultat est imprimé par la ligne 3ØØ, le changement de joueur est réalisé par la ligne 31Ø et un "." est imprimé à l'emplacement de départ. Le sous-programme 33Ø-34Ø est utilisé pour les divers contrôles.

POUR JOUER :
On lance le programme par RUN. Le terrain de jeu est affiché, ainsi que le symbole permettant d'identifier le joueur qui a le trait. Le joueur peut choisir le pion qu'il désire déplacer en manœuvrant le curseur clignotant grâce aux touches de déplacement du curseur.
Pour déplacer son pion, il suffit, lorsque le curseur est sur ce pion, de taper la direction du déplacement, suivant le code :

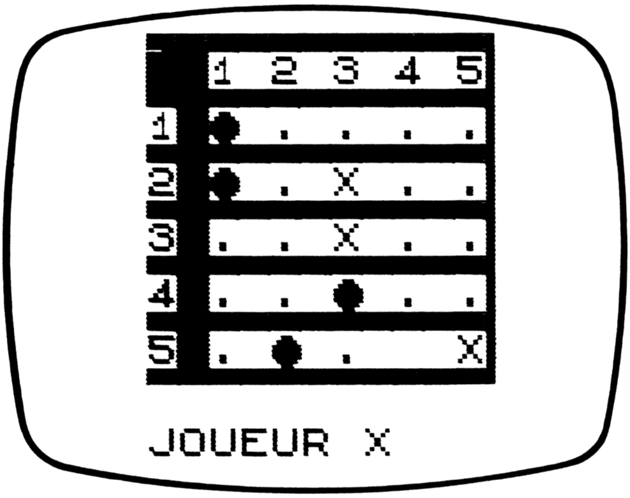


Le coup sera refusé si la case de départ n'est pas conforme ou si la case d'arrivée est occupée. Sinon, le coup est réalisé, et le trait passe à l'autre joueur, qui déplace le curseur clignotant pour l'amener sur l'un des pions, et joue de la même façon.
Le joueur gagnant est celui qui réussit à amener le premier tous ses pions dans les cases initialement occupées par l'adversaire. On peut interrompre la partie à tout moment en tapant CNT et C.

EXTENSIONS POSSIBLES :
On peut améliorer les contrôles, prévoir un test de fin, utiliser un terrain de jeu différent ; on pourra aussi envisager l'écriture d'un programme permettant au MO6 de participer au jeu avec de bonnes chances de l'emporter.

```
10 REM QUATRE A QUATRE
20 CLS:SCREEN 4,6:CLEAR,,1:DEFGR$(0)=0,2
8,62,127,127,127,62,28
30 BOXF(0,0)-(180,150),0
40 H=1:A$="5665322356"
50 ATTRB 1,1:COLOR 1,2:LOCATE 4,2:PRINT"
1 2 3 4 5"
60 FOR I=1 TO 5
70 LOCATE 0,3*I+2:COLOR 1:PRINT RIGHT$(S
TR$(I),1):LOCATE 4,3*I+2:PRINT". . . . .
"
80 IF I<3 THEN LOCATE 4,3*I+2:COLOR 4,2:
PRINT GR$(0)+" "+GR$(0)
90 IF I>3 THEN LOCATE 16,3*I+2:COLOR 3,2
:PRINT"X X"
100 NEXT I
110 X=1:Y=1
120 LOCATE 0,22:COLOR 4+(H=0),6:IF H=1 T
HEN PRINT"JOUEUR ";GR$(0) ELSE PRINT "JO
UEUR X"
130 GOSUB 330
140 I=J
150 LOCATE 4*X,3*Y+2,0:COLOR ,2:PRINT "
"
160 A=X:B=Y
170 R$=INKEY$
180 Y=Y - (R$=CHR$(10) AND Y<5) + (R$=CH
R$(11) AND Y>1)
190 X=X - (R$=CHR$(9) AND X<5) + (R$=CHR
$(8) AND X>1)
200 IF R$<"9" AND R$>"0" THEN 230
210 LOCATE 4*A,3*B+2,0:COLOR J-4*(J=-3),
2:IF J=4 THEN PRINT GR$(0) ELSE IF J=3 T
HEN PRINT"X"ELSE PRINT"."
220 GOTO 120
230 IF (I=-3) OR (I=3 AND H=1) OR (I=4 A
ND H=0) THEN 210
240 J=VAL(R$)
250 S=Y:Y=Y + VAL(MID$(A$,J,1))-4
```

```
260 T=X:X=X + VAL(MID$(A$,J+2,1))-4
270 IF X<1 OR X>5 OR Y<1 OR Y>5 THEN Y=S
:X=T:A=T:B=S:J=3-(H=1):GOTO 210
280 GOSUB 330
290 IF J<>-3 THEN A=T:B=S:J=3-(H=1):GOTO 320
300 LOCATE 4*X,3*Y+2,0:COLOR 4+(H=0),2:I
F H=0 THEN PRINT"X" ELSE PRINT GR$(0)
310 H=ABS(H-1):I=1:PLAY"DOREMI"
320 GOTO 210
330 J=POINT(32*X+8,24*Y+16)
340 RETURN
```



Echecs

REFERENCES :
Difficulté de programmation : difficile.
Catégorie de jeu : jeu de réflexion, pour deux joueurs.
Difficulté du jeu : difficile.

LE JEU :
Le programme propose seulement à deux joueurs de disputer une partie en se servant de l'écran du téléviseur comme plateau de jeu. Il n'y a pas de contrôle de la validité des coups, mais seulement gestion de la position des pièces sur l'échiquier. Les pions et les pièces sont dessinés à l'aide des caractères graphiques GR$. La couleur permet de distinguer chaque camp.

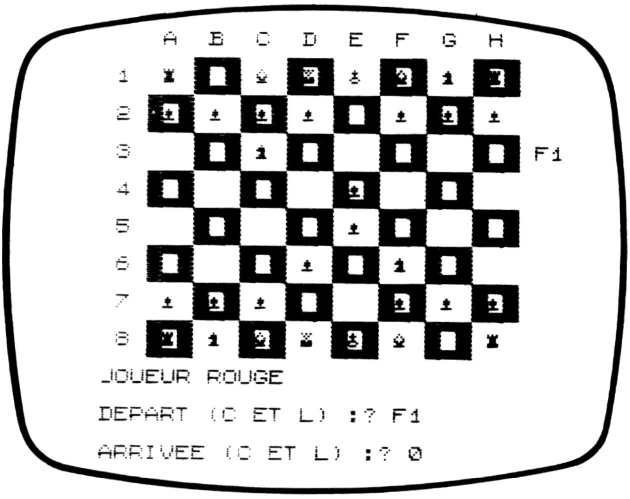
LE PROGRAMME :
Les lignes 100 à 200 dessinent l'échiquier. Les lignes 210 et 220 définissent les conditions initiales, à l'aide de deux tableaux : R (I, J) pour les codes des pièces, et C (I, J) pour la couleur.
Les boucles emboîtées J (240-290) et I (250-280) dessinent les pièces.
La ligne 320 permet l'entrée de la ligne et de la colonne de départ, avec contrôle en ligne 350.
La ligne 370 permet d'entrer colonne et ligne d'arrivée.
La ligne 400 note la nouvelle position.
La ligne 410 assure le changement de joueur.

POUR JOUER :
On lance l'exécution par RUN.
L'échiquier est dessiné, puis les pièces, et enfin l'inscription JOUEUR BLEU suivie de DEPART (C et L) ; on entre la colonne et la ligne de départ, sous la forme E 7 par exemple, on obtient alors l'affichage de ARRIVEE (C et L) et on répond de même. Le déplacement est alors effectué, et c'est au tour de l'autre joueur.
Pour arrêter, il suffit de taper CNT et C.

EXTENSIONS POSSIBLES :
On pourra chercher à réaliser un graphisme plus précis, introduire le contrôle du mouvement des pièces. Et pourquoi pas, écrire un programme qui permette au MO6 de se comporter en partenaire honorable. Bon courage !

```
10 REM ECHECS
20 DIM R(8,8):DIM C(8,8):CLEAR,,9:DEFGR$
(6)=0,0,0,0,0,0,0,0:DEFGR$(7)=0,0,0,0,25
5,255,255,255:DEFGR$(8)=255,255,255,255,
0,0,0,0
30 DEFGR$(0)=0,16,40,68,108,56,124,0:DEF
GR$(1)=0,16,56,16,56,68,56,0
40 DEFGR$(2)=0,84,124,56,56,124,124,0:DE
FGR$(3)=0,84,40,16,108,124,124,0
50 DEFGR$(4)=0,0,16,56,56,16,124,0:DEFGR
$(5)=0,16,56,120,24,56,124,0
60 A$=GR$(7)+GR$(7)+GR$(7)
70 B$=CHR$(127)+CHR$(127)+CHR$(127)
80 C$=GR$(8)+GR$(8)+GR$(8)
90 CLS:SCREEN 0,7
100 PRINT"    A  B  C  D  E  F  G  H"
110 FOR I=1 TO 8
120 FOR J=1 TO 8
130 IF (I+J)/2=INT((I+J)/2) THEN 150
140 LOCATE 3*J,2*I-1:PRINT A$:PRINT I:LO
CATE 3*J,2*I:PRINT B$:LOCATE 3*J,2*I+1:P
RINT C$
150 IF I=2 OR I=7 THEN R(I,J)=4
160 IF I<>2 AND I<>7 THEN R(I,J)=6
170 C(I,J)=1
180 IF I=7 OR I=8 THEN C(I,J)=6
190 NEXT J
200 NEXT I
210 R(1,1)=2:R(1,2)=5:R(1,3)=0:R(1,4)=3:
R(1,5)=1:R(1,6)=0:R(1,7)=5:R(1,8)=2
220 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
230 H=1
240 FOR J=1 TO 8
250 FOR I=1 TO 8
260 X=1+3*J:Y=2*I:F=POINT(8*X+10,8*Y+10)
:IF F=-8 THEN F=7
270 LOCATE X,Y,0:COLOR C(I,J),F:PRINT GR
$(R(I,J))
280 NEXT I
290 NEXT J
300 LOCATE 28,6:COLOR 0,7:PRINT"       
":LOCATE 18,20:PRINT"            ":LOCATE 19,22:
PRINT"    "
310 LOCATE 0,18:IF H=1 THEN PRINT"JOUEUR
 BLEU " ELSE PRINT"JOUEUR ROUGE"
320 LOCATE 0,20:INPUT"DEPART (C ET L) :"
;D$
330 LOCATE 28,6:PRINT D$
340 GOSUB 500
350 IF R(L,C)=6 OR (C(L,C)=1 AND H=1) OR
 (C(L,C)=6 AND H=0) THEN 310
360 T=R(L,C):S=C(L,C):R(L,C)=6
370 LOCATE 0,22:INPUT"ARRIVEE (C ET L) :
```

```
";D$
380 LOCATE 31,6:PRINT "- ";D$
390 GOSUB 500
400 R(L,C)=T:C(L,C)=S
410 H=ABS(H-1)
420 GOTO 240
500 L=VAL(RIGHT$(D$,1)):C=ASC(LEFT$(D$,1
))-64
510 RETURN
```



Achevé d'imprimer par l'imprimerie Tardy Quercy S.A. - Bourges
D.L. : Décembre 1986 - N° d'Imprimeur : 13373 - N° d'Édition : 86595-393-1

Votre avis nous intéresse

Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent ouvrage.

— Ce livre vous donne-t-il toute satisfaction ?

....
....
....
....
....

— Y a-t-il un aspect du problème que vous auriez aimé voir abordé ?

....
....
....
....
....

Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous ne manquerons pas de vous répondre directement.

Où avez-vous acheté ce livre ?

☐ cadeau ☐ librairie ☐ autres
☐ exposition ☐ boutique micro

Comment en avez-vous eu connaissance ?

☐ publicité ☐ catalogue ☐ autres
☐ exposition ☐ conseils d'un ami

Avez-vous déjà acquis des livres P.S.I. ?

Lesquels ? ....
....
qu'en pensez-vous ? ....
....
....

Nom .... Prénom .... Age........
Adresse ....
Profession ....
Centre d'intérêt ....

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
5, place du Colonel-Fabien
75491 PARIS CEDEX 10

Apprendre en se distrayant, tel est l'objectif de ce livre.
Au fil de ces 102 programmes de jeux il vous guidera dans l'exploration du Basic des M06, T08 et T09+. Les programmes sont classés par niveau, chacun d'eux faisant appel à de nouvelles connaissances et à une plus grande maîtrise du Basic.
Chaque niveau commence par une présentation concise de nouvelles instructions utilisées.
Tous les jeux sont décrits, et les programmes abondamment commentés;
un exemple d'exécution est fourni pour chaque version.
Ce livre est la nouvelle édition de
102 programmes pour M06, T08 et T09+.

DES LIVRES
POUR VOTRE
ORDINATEUR

P.S.I.

ISBN : 2.86595.393.9/135 FF

9 782865 953936

Couverture GIBRAT

102 PROGRAMMES POUR MO6, TO8 ET TO9+

PAR JACQUES DECONCHAT

PROGRAMMES

P.S.I.